Self-Validated Labeling of Markov Random Fields for Image Segmentation

Wei Feng, Jiaya Jia, *Member*, *IEEE*, and Zhi-Qiang Liu

Abstract—This paper addresses the problem of *self-validated labeling* of Markov random fields (MRFs), namely to optimize an MRF with unknown number of labels. We present *graduated graph cuts* (GGC), a new technique that extends the binary s-t graph cut for self-validated labeling. Specifically, we use the split-and-merge strategy to decompose the complex problem to a series of tractable subproblems. In terms of Gibbs energy minimization, a suboptimal labeling is gradually obtained based upon a set of cluster-level operations. By using different optimization structures, we propose three practical algorithms: tree-structured graph cuts (TSGC), net-structured graph cuts (NSGC), and hierarchical graph cuts (HGC). In contrast to previous methods, the proposed algorithms can automatically determine the number of labels, properly balance the labeling accuracy, spatial coherence, and the labeling cost (i.e., the number of labels), and are computationally efficient, independent to initialization, and able to converge to good local minima of the objective energy function. We apply the proposed algorithms to natural image segmentation. Experimental results show that our algorithms produce generally feasible segmentations for benchmark data sets, and outperform alternative methods in terms of robustness to noise, speed, and preservation of soft boundaries.

Index Terms—Self-validated labeling, Markov random fields (MRFs), graduated graph cuts, image segmentation, split-and-merge.

1 Introduction

As an intermediate process, image segmentation plays an important role in many high-level vision tasks, such as recognition [1], [2], [3] and authentication [4]. Recently, many methods have been proposed for the K-labeling segmentation, which mainly concerns the balance of segmentation quality and computational efficiency, under the assumption that the number of labels K is known [5], [6], [7], [8], [9], [10]. However, this assumption makes them highly unsuitable for recent content-based applications, such as generic object recognition [11], since, for a large number of images, it is very infeasible to ask the users to input the "right" number of segments for each individual image. Therefore, in image segmentation, we should consider not only the labeling quality (i.e., the Gibbs energy integrating labeling accuracy and spatial coherence) but also the labeling cost (i.e., the number of labels). In general, for a comparable segmentation quality (i.e., similar Gibbs energy value), we should prefer the labeling with the least number of labels due to Occam's razor. For clarity, we use the term *self-validated labeling* to denote the labeling problem when the number of labels is unknown.[1]

Despite the practical importance, self-validated labeling is far from being well-studied for image segmentation. There seems a straightforward way to extend the K-labeling segmentation to be self-validated—that is, we try several particular K and choose the best labeling result with the smallest K. However, this is highly impractical, especially for a large number of images, where complexity is the most obstacle (see Section 4.4 for details). In addition, some K-labeling methods may lead to unsatisfactory segmentations, even when fed by the "correct" number of labels. See Fig. 1b for example. For the synthesized image in Fig. 1a, normalized cut [6], a state-of-the-art K-labeling method, is unable to generate satisfying segmentation using the correct number of labels, 5. It can produce all "right" boundaries, together with many false-alarmed boundaries, only by feeding with a much larger number of labels, 20. Indeed, the drawback of K-labeling segmentation emphasizes the practical necessity of self-validated labeling.

As shown in Fig. 1c, there exist some self-validated methods in the literature [13], [14], [15], [16]. However, few of them succeed in both computational efficiency and robustness to noise distortion. For instance, split-and-merge is a natural strategy for self-validated segmentation (see GBS in Fig. 1c for example) and has a long history [17]. In general, existing split-and-merge methods are able to efficiently determine all salient regions of an image by coarse-to-fine partitioning (e.g., IsoCut [18]) and/or fine-to-coarse aggregation (e.g., SWA [19]). By using advanced optimization techniques, such as spectral relaxation in SWA, recent split-and-merge methods have largely improved their accuracy and flexibility. More importantly,

- W. Feng is with the School of Creative Media, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong, S.A.R., China, and was with The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong, S.A.R., China. E-mail: wfeng@ieee.org.
- J. Jia is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong, S.A.R., China. E-mail: leojia@cse.cuhk.edu.hk.
- Z.-Q. Liu is with the School of Creative Media, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong, S.A.R., China. E-mail: zq.liu@cityu.edu.hk.

Manuscript received 5 Sept. 2008; revised 9 June 2009; accepted 25 Sept. 2009; published online 15 Jan. 2010.
Recommended for acceptance by R. Ambasamudram.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-09-0598.
Digital Object Identifier no. 10.1109/TPAMI.2010.24.

[1] The term self-validated labeling (or self-validation) is borrowed from clustering validation [12], where the number of clusters is a very important aspect to be validated.

0162-8828/10/$26.00 © 2010 IEEE Published by the IEEE Computer Society

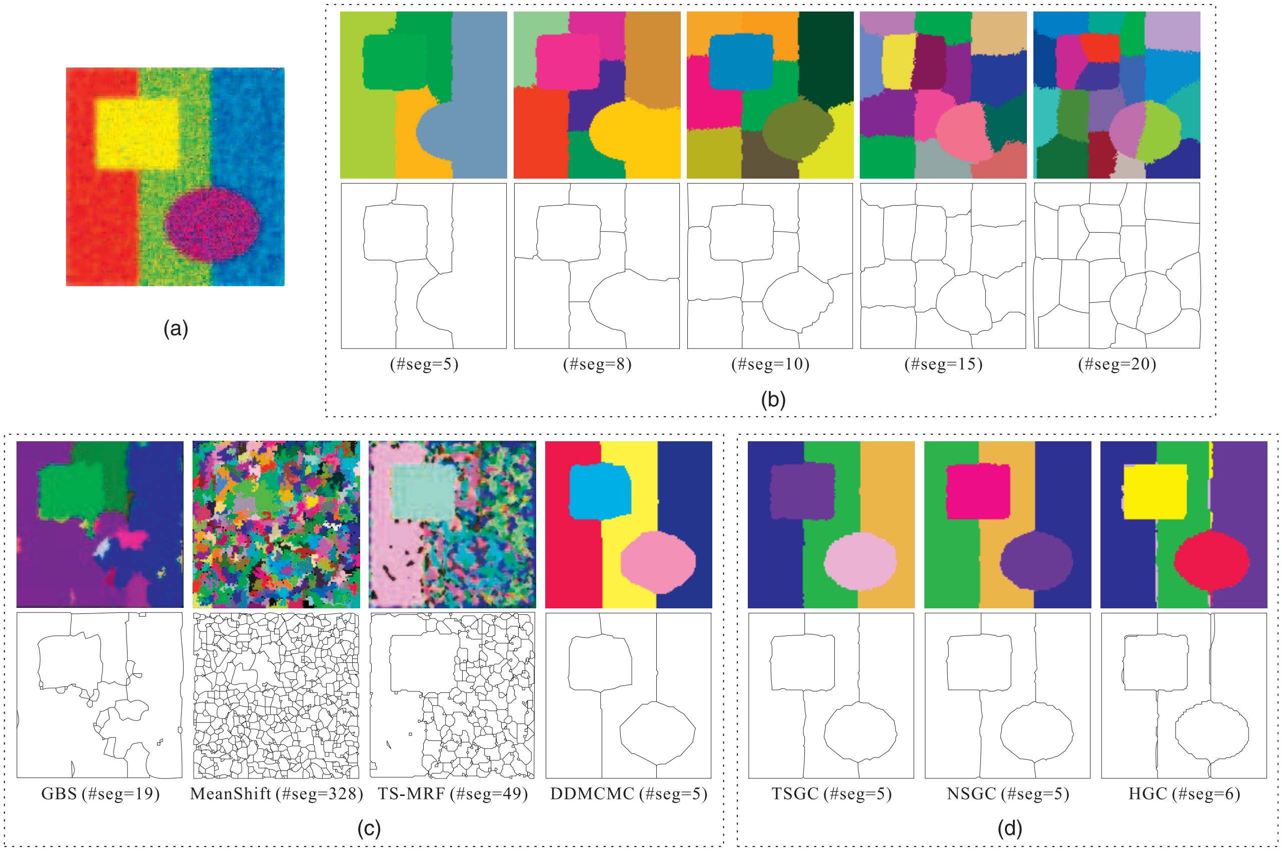


Fig. 1. Comparison of self-validated and K-labeling segmentation for the synthesized image (a) (of size 128×128, consisting of 5 segments and with 40 percent Gaussian noise). We show the segmentation results of normalized cut (NCut) [6], which is a representative K-labeling method, and four state-of-the-art self-validated methods (efficient graph-based segmentation (GBS) [13], MeanShift [14], tree-structured MRF (TS-MRF) [15], and data-driven Markov Chain Monte Carlo (DDMCMC) [16]) in (b) and (c), respectively. The average running time of NCut for image (a) is 11.4 s. However, NCut is unable to return satisfying result when fed by the correct number of segments, 5; it can produce all "right" boundaries, mixed with many "wrong" boundaries, only when fed by a much larger number of segments, 20. Bottom-up methods, such as MeanShift and GBS, can automatically determine the number of segments and are very fast (need about 1 s), but are either apt to oversegmentation or too sensitive to noise corruption. TS-MRF uses split-and-merge strategy in the MRF framework, but the approximate optimization cannot guarantee spatially coherent labeling. DDMCMC generates a reasonable result with the right number of segments, but it is very slow and needs about 210 s to reach a stable labeling for image (a). The results of the proposed algorithms are shown in (d). Note that TSGC needs 2.96 s, NSGC needs 5.70 s, HGC needs 2.01 s, and all of them produce satisfying results with reasonable number of segments. (a) Original. (b) K-labeling segmentation (normalized cut). (c) Self-validated segmentation (previous methods). (d) Self-validated segmentation (our methods).

most of them are self-validated and efficient. However, these methods are usually not robust enough to noise distortion. In contrast, K-labeling methods are accurate, robust, but not self-validated. Therefore, a reliable labeling method which can automatically determine the number of labels and optimize a global objective function is of significant importance both theoretically and practically.

In this paper, we address this problem and study how to "suboptimally" label an MRF with unknown number of labels. Our work aims at seeking a simple yet principled way to decompose the intractable self-validated labeling problem to a series of tractable subproblems. We start from the graph theoretic formulation of image segmentation. It is Greig et al. who first showed the binary labeling problem can be exactly solved by s-t graph mincut/maxflow (s-t graph cut for short) [20]. Recently, Boykov et al. horizontally extended the binary s-t graph cut to solve the K-labeling problem [21], [22], [23], which was further theoretically investigated by Kolmogorov et al. [24], [25]. Except for being not self-validated, the horizontal extension, as shown by our experiments, is highly dependent on the initialization. We propose an alternative way to vertically extend the binary s-t graph cut to solve the self-validated labeling problem.[2] The core of our approach is to gradually reconstruct a suboptimal segmentation based on a set of cluster-level operations: *retaining*, *splitting*, *merging*, and *regrouping*. These operations lead to a suboptimal top-down labeling refinement process which intrinsically solves the self-validation problem of MRFs labeling. Because we gradually refine the labeling using s-t graph cut, we name our approach *graduated graph cuts* (GGC).

Using different optimization structures, we propose three concrete algorithms: tree-structured graph cuts (TSGC), net-structured graph cuts (NSGC), and hierarchical graph cuts (HGC), within the GGC scheme. All of them start by treating the whole image as a single segment, then iteratively select the optimal operation for each segment,

[2] The term vertical/horizontal extension is derived from the algorithmic structure of segmentation methods, which refers to whether a segmentation algorithm explores the space of all possible labelings directly or in a hierarchical manner. As shown in Fig. 6, the proposed GGC scheme uses the s-t graph cut to gradually refine the labeling between two consecutive levels in the hierarchical structure and thus can be viewed as a vertical extension to the binary s-t graph cut. In contrast, α-expansion and α-β-swap use the s-t graph cut to iteratively refine the K-labeling at the same level, thus belonging to horizontal extension.

and end when the overall energy stops decreasing. Based on extensive experiments and a qualitative analysis, we show that, by this process, the proposed algorithms always obtain good local minima of the objective energy function.

2 Related Work

Image segmentation remains a classical and active topic in low-level vision for decades. Among the previous successful methods, MRF-based ones account for a large percentage [26]. In recent years, it is still a dynamic area that studies the application of MRFs in image segmentation, such as double MRF [27], data-driven MCMC [16], hidden Markov measure field model [28], and tree-structured MRF [15]. MRF-based methods can naturally represent the labeling accuracy and spatial coherence within the Bayesian framework, but some of them are computationally complex and converge slowly.

Recently, graph theoretic approaches have been successfully used in MRF labeling [5], [6], [7], [8], [9], [18], [29]. These methods treat image segmentation as a graph partitioning problem. By using advanced optimization techniques, they realize efficient energy minimization and properly balance the labeling quality and computational efficiency. According to the particular energy function they minimize, graph theoretic approaches fall into two categories: continuous methods and discrete methods.

Representative continuous segmentation methods include spectral relaxation [6], [30], [31], [32], [33] and random walk [7], [8], [9]. By encoding local pairwise similarity into a graph, continuous methods usually minimize a relaxed partitioning function by solving a constrained eigen-problem [6] or a linear system [9]. Since continuous methods generate fractional labeling, they were successfully used in image matting [34], [35]. In contrast, typical discrete methods, such as s-t graph cut [20] and its extensions [21], [22], [23], [24], [25], directly solve the original integer-programming for graph partitioning. One desirable property of discrete methods is that, for a particular set of binary energy functions (e.g., submodular functions [24], [25]), they can obtain a globally optimal labeling in polynomial time. Since in image segmentation the widely used pairwise terms based on the Potts model are usually submodular, discrete segmentation methods usually have a better theoretical guarantee, especially in the binary case. For instance, binary s-t graph cut has achieved great success in interactive foreground segmentation [36], [37], [38]. Although discrete methods guarantee global optima for some particular energy forms, they cannot handle generic energy functions [39], while continuous methods have no such limit. Moreover, continuous MRF models have recently shown their potential in rectifying the shrinkage bias of discrete models in seeded image segmentation [7], [8]. Despite the respective merits of discrete and continuous methods in image segmentation, how to properly take account of the labeling cost (i.e., the number of labels) within the optimization framework (for either discrete or continuous models) still remains an open problem.

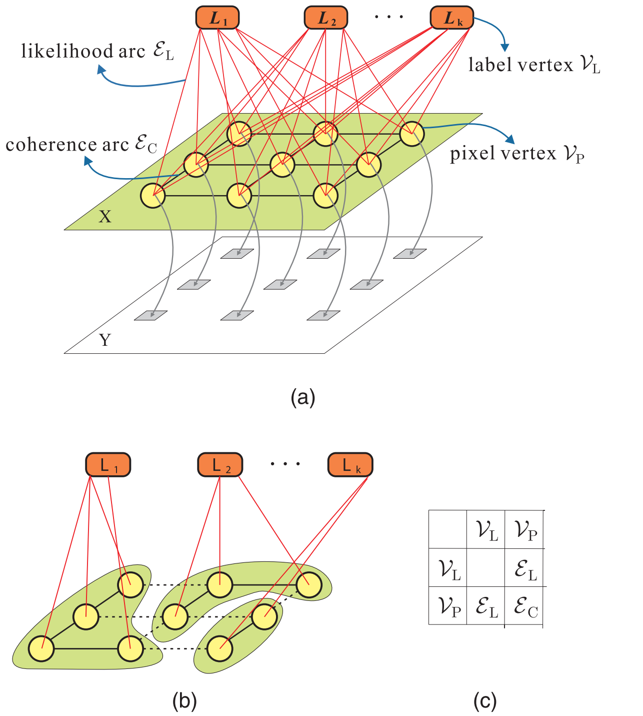
Before image segmentation, graph partitioning was widely used in parallel simulation and the design of VLSI. There are some well-established graph partitioning packages, such as METIS [40], which use multilevel split-and-merge to seek a balanced minimal partition to the graph using the Kernighan-Lin algorithm. Despite their successes in other areas, they are not readily suitable for image segmentation since they cannot maintain spatial coherence well, are not guaranteed to be optimal, and are biased for image segmentation. For instance, the normalized cut criterion was proposed to rectify the small partition bias of minimum cut criterion for image segmentation [6]. The improvement can be clearly seen in Fig. 16. But, the split-and-merge strategy does shed light on self-validated labeling since many split-and-merge segmentation methods are self-validated [13], [18]. Note that there is a long history of split-and-merge in image segmentation [17]. Recent methods, such as IsoCut [18] and GBS [13], integrate advanced optimization techniques into the split-and-merge scheme, thus significantly improving the performance. However, as shown by our experiments (see Figs. 12 and 13), they are usually sensitive to noise distortion.

In this paper, we try to find a principled way to apply the split-and-merge strategy to a global optimization framework, thus integrating the merits of both schemes and realizing reliable self-validated labeling. In contrast to Boykov's horizontal extension to binary s-t graph cut by iteratively refining the labeling between the K segments [21], our GGC approach can be viewed as a vertical extension. We show how to gradually use the binary s-t graph cut to efficiently minimize the Gibbs energy downward, thus solving the self-validated labeling problem. Our work is also partially related to clustering, e.g., AutoClass [41] uses a mixture prior model and the Bayesian method to determine the optimal classes, which intrinsically favors clustering with less numbers of clusters. Another related work is TS-MRF [15], which reduces the K-labeling segmentation into a sequence of $K-1$ simpler binary segmentations. However, the suboptimality property of TS-MRF is unclear, and as shown in our experiments, the approximate labeling of TS-MRF may arbitrarily deviate from good local minima, thus severely degrading the performance. Partial work and some preliminary results of this paper were also reported in [29].

3 Graph Formulation for Segmentation

The Bayesian framework and MRFs model form a solid foundation for image segmentation, based on which an image I is modeled as an MRF X with observation Y. The MRF $X = \{x_p\}$ represents the segmentation result that considers both accuracy and contextual constraints, where $x_p \in \mathcal{L}$, p is an image pixel, and $\mathcal{L} = \{L_1, L_2, \ldots\}$ is the label space. The goal of self-validated segmentation is to estimate the optimal label for each pixel when the number of labels $K = |\mathcal{L}|$ is unknown.

As shown in Fig. 2a, we can equivalently interpret the MRF model with a graph formulation, where the image I and the contextual dependency is represented by an undirected graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$. In particular, the vertex set $\mathcal{V} = \{\mathcal{V}_{\rm L}, \mathcal{V}_{\rm P}\}$ is composed of two components: the set of label vertices $\mathcal{V}_{\rm L}$ and the set of pixel vertices $\mathcal{V}_{\rm P}$. From Fig. 2, we can see that each v_p ($v_p \in \mathcal{V}_{\rm P}$) corresponds to a pixel p, with y_p denoting its observation and x_p ($x_p \in \mathcal{V}_{\rm L}$) denoting its potential label. The arc set $\mathcal{E} = \{\mathcal{E}_{\rm L}, \mathcal{E}_{\rm C}\}$ also consists of two parts: the likelihood arcs $\mathcal{E}_{\rm L} = \{(v_l, v_p) | v_l \in \mathcal{V}_{\rm L} \text{ and } v_p \in \mathcal{V}_{\rm P}\}$ representing the likelihood of labeling pixel p with v_l ($v_l = L_l$), and the coherence arcs $\mathcal{E}_{\rm C} = \{(v_p, v_q) | (v_p, v_q) \in \mathcal{V}_{\rm P}^2 \text{ and } (p, q) \in \mathcal{N}_2\}$ reflecting the pairwise dependency between adjacent pixels. Here $\mathcal{N}_2$ denotes the second order neighborhood system. Although the proposed approach


L1
L2
. . .
Lk
likelihood arc $\mathcal{E}_\mathrm{L}$
label vertex $\mathcal{V}_\mathrm{L}$
coherence arc $\mathcal{E}_\mathrm{C}$
pixel vertex $\mathcal{V}_\mathrm{P}$
X
Y
L1
L2
. . .
Lk

(a)

	$\mathcal{V}_\mathrm{L}$	$\mathcal{V}_\mathrm{P}$
$\mathcal{V}_\mathrm{L}$		$\mathcal{E}_\mathrm{L}$
$\mathcal{V}_\mathrm{P}$	$\mathcal{E}_\mathrm{L}$	$\mathcal{E}_\mathrm{C}$

(b) (c)

Fig. 2. Graph formulation of MRF-based segmentation (by 4-connected lattice): (a) undirected graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ with K segments $\{L_1, \dots L_K\}$ and the observation Y; (b) final segmentation corresponds to a multiway cut of the graph $\mathcal{G}$; (c) the relationship of vertices and edges in the graph formulation.

can handle dense graphs with long-range pairwise connections of any type, we focus on the 4-connected lattice in this paper. Note that, in MRFs labeling, it has been shown that properly encoding long-range connections helps to improve the labeling specificity [42], [43], [44]. In image segmentation, this means that dense graphs with long-range connections may benefit the preservation of soft boundaries. However, the cost is a significant increase to the complexity of graph partitioning. In contrast, we show in Fig. 14 that our approach is good at preserving soft boundaries using the simple 4-connected lattice only, which results in high computational efficiency.

In terms of MRFs, the statistically optimal labeling $\hat{X}$ maximizes the *posteriori* probability (MAP) given observation Y, i.e., $\hat{X} = \arg\max_X P(X|Y)$, which is equivalent to minimizing the following Gibbs energy:

$$E(X \mid Y) = \sum_{(x_p,\, v_p) \in \mathcal{E}_\mathrm{L}} E_{\mathrm{lik}}(x_p \mid y_p) \; + \sum_{(v_p,\, v_q) \in \mathcal{E}_\mathrm{C}} E_{\mathrm{coh}}(x_p,\ x_q), \qquad (1)$$

where $E_{\mathrm{lik}}(x_p)$ is the likelihood energy representing the goodness of labeling pixel p by x_p and $E_{\mathrm{coh}}(x_p, x_q)$ is the coherence energy denoting the prior of labeling spatial coherence. For simplicity, in the rest of the paper, we use $E(X)$ to represent the Gibbs energy $E(X|Y)$, use $E_{\mathrm{lik}}(e_p)$ ($e_p = \{x_p, v_p\} \in \mathcal{E}_\mathrm{L}$) to represent $E_{\mathrm{lik}}(x_p|y_p)$, and use $E_{\mathrm{coh}}(e_{pq})$ ($e_{pq} = \{v_p, v_q\} \in \mathcal{E}_\mathrm{C}$) to represent $E_{\mathrm{coh}}(x_p, x_q)$. Then, we can rewrite (1) as

$$E(X) = \sum_{e_p \in \mathcal{E}_\mathrm{L}} E_{\mathrm{lik}}(e_p) \; + \sum_{e_{pq} \in \mathcal{E}_\mathrm{C}} E_{\mathrm{coh}}(e_{pq}). \qquad (2)$$

Referring to Fig. 2a, if we assign $E_{\mathrm{lik}}(e_p)$ as the weight of likelihood arc e_p and $E_{\mathrm{coh}}(e_{pq})$ as the weight of coherence arc e_{pq}, the optimal segmentation with minimal Gibbs energy corresponds to the minimal multiway cut of weighted graph $\mathcal{G}$ (see Fig. 2b).

Let $\mathrm{Seg}(I) = \{S_i\}_{i=1}^K$ represent the final segmentation of image I consisting of K nonoverlapping segments $S_1, \dots, S_K$, each of which has a unique label $L(S_i) \in \mathcal{L}$ and $L(S_i) \neq L(S_j)$ when $i \neq j$. A particular segment S corresponds to a set of pixels having the same label, i.e., $S = \{v | v \in \mathcal{V}_\mathrm{P} \text{ and } L(v) = L(S)\}$, where $L(v)$ and $L(S)$ denote the labels of pixel v and segment S, respectively. We use $F(S)$ to represent the corresponding feature space of S, i.e., $F(S) = \{y_p | v_p \in S\}$. From (1) and (2), we can express the likelihood and coherence energy functions of segment S as

$$E_{\mathrm{lik}}(S) = \sum_{\substack{(L(S),\, v_p) \in \mathcal{E}_\mathrm{L} \\ v_p \in S}} E_{\mathrm{lik}}(L(S) \mid y_p), \qquad (3)$$

$$E_{\mathrm{coh}}(S) = \sum_{\substack{(v_p,\, v_q) \in \mathcal{E}_\mathrm{C} \\ v_p \in S \,\wedge\, v_q \notin S}} E_{\mathrm{coh}}(x_p,\ x_q). \qquad (4)$$

Specifically, the coherence energy of segment pair (S_i, S_j) ($S_i \neq S_j$) is

$$E_{\mathrm{coh}}(S_i, S_j) = \sum_{\substack{(v_p,\, v_q) \in \mathcal{E}_\mathrm{C} \\ v_p \in S_i \,\wedge\, v_q \in S_j}} E_{\mathrm{coh}}(x_p,\ x_q). \qquad (5)$$

It is clear that $E_{\mathrm{coh}}(S_i) = \sum_{j \neq i} E_{\mathrm{coh}}(S_i, S_j)$. Based on (1)-(5), we can express the Gibbs energy of segmentation $\mathrm{Seg}(I)$ as

$$\begin{aligned} E\big(\mathrm{Seg}(I)\big) &= \sum_{k=1}^{K} E_{\mathrm{lik}}(S_k) + \sum_{i,j=1}^{K} E_{\mathrm{coh}}(S_i, S_j) \\ &= \sum_{k=1}^{K} \left(E_{\mathrm{lik}}(S_k) + \frac{E_{\mathrm{coh}}(S_k)}{2} \right). \end{aligned} \qquad (6)$$

Graph formulation reveals the equivalence between MRF-based segmentation and the NP-complete multiway (K-way) graph cut problem [21], [45]. Furthermore, when the number of labels K is unknown, the problem becomes even more difficult. Nevertheless, for $K = 2$, it is equivalent to the classical graph mincut/maxflow problem, which can be exactly minimized in polynomial time [21], [22], [23], [24], [46].

3.1 Optimal Binary Segmentation

Let us first see how to use the s-t graph cut to achieve an optimal binary segmentation, i.e., labeling all pixels with a binary MRF $X^{\mathrm{b}} = \{x_p\}$ ($x_p \in \{0, 1\}$) based on observation Y. When $K = 2$, the Gibbs energy of segmentation becomes

$$E^{\mathrm{b}}(X) = \sum_{e_p \in \mathcal{E}_\mathrm{L}} E^{\mathrm{b}}_{\mathrm{lik}}(e_p) \; + \sum_{e_{pq} \in \mathcal{E}_\mathrm{C}} E^{\mathrm{b}}_{\mathrm{coh}}(e_{pq}). \qquad (7)$$

As we have known, a particular difficulty in segmentation is that one has to estimate both the representative model of each segment and its corresponding regions at the same time [28]. Thus, we need an effective feature model to represent each segment, and based on which to define the concrete energy function of $E^{\mathrm{b}}_{\mathrm{lik}}$ and $E^{\mathrm{b}}_{\mathrm{coh}}$.

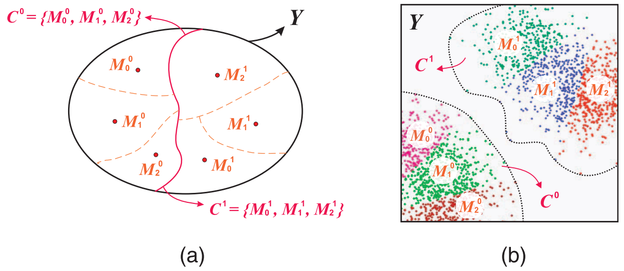


Fig. 3. Nonparametric feature model for a segment with observation Y ($H = 3$): (a) is the two-level component/subcomponent representation, (b) shows an example.

3.1.1 Feature Space Representation

To achieve an effective energy assignment and efficient computation, we use Algorithm 1 to derive a two-level (*component/subcomponent*) nonparametric feature model for a segment with observation feature space Y (see Fig. 3 for illustration). Specifically, for a given segment S, its feature space is characterized by 2 components (used to compute the optimal segment splitting energy) and $2H$ subcomponents (used to compute the segment retaining and merging energies). Details about their functions can be found in (8)-(11) and Section 4.

```
Algorithm 1. Two-level nonparametric feature model
  Input: Feature space $Y$.
  Output: Two-level component/subcomponent feature
          model $\{C^0, C^1\} = \{\{M_h^0\}_{h=1}^H, \{M_h^1\}_{h=1}^H\}$.
  1. Dividing feature samples into two components $C^0$ and
     $C^1$ by K-means;
  2. Subdividing $C^0$ and $C^1$ into $H$ subcomponents $M_h^0$ and
     $M_h^1$ by K-means, respectively;
  3. Deriving the representative center for each
     subcomponent and obtaining the final model
     $Y = \{C^0, C^1\} = \{\{M_h^0\}_{h=1}^H, \{M_h^1\}_{h=1}^H\}$;
```

Note that our feature model is nonparametric, which is in contrast to the GMM modeling in GrabCut [38]. We choose the nonparametric manner mainly due to the consideration of efficiency. As we know, the EM estimation of GMM model needs an iterative process. It is unsuitable for self-validated segmentation since we may need to model a large number of segments, as opposed to the two segments in GrabCut. Instead, we use a fast K-means algorithm in our scheme with triangle inequality acceleration [47]. Besides, the nonparametric representation helps to improve the ability of our model to handle non-Gaussian color distributions in an image.

3.1.2 Energy Assignment

The likelihood energy measures the segmentation accuracy and can be viewed as the intrasegment distance. Based on the nonparametric representation of feature space, we define the likelihood energy as

$$E_{\text{lik}}^{\text{b}}(x_p) = \left(d_p^1 x_p + d_p^0(1 - x_p)\right)^{\alpha}, \tag{8}$$

where $d_p^0 = D(y_p, C^0)$ and $d_p^1 = D(y_p, C^1)$ represent the distance between y_p and C^0 and C^1, respectively, and

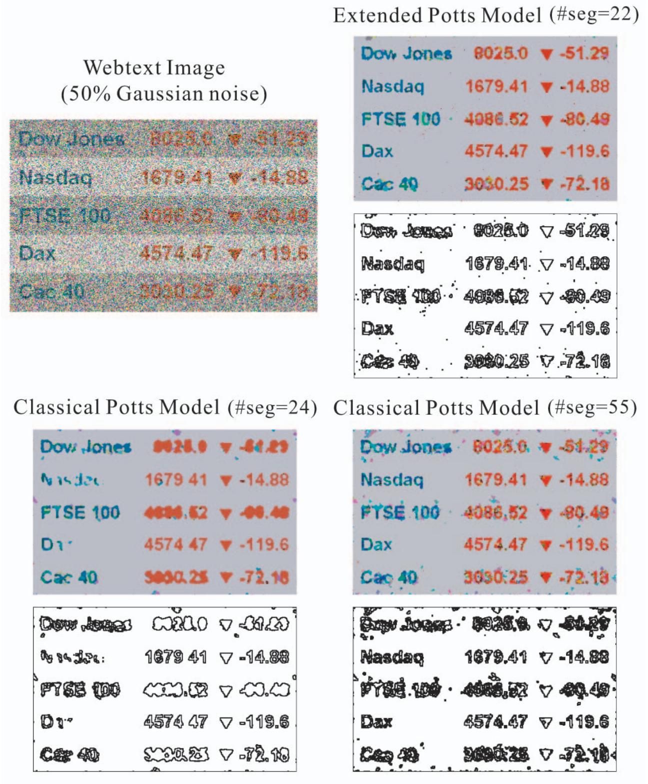

Webtext Image
(50% Gaussian noise)
Extended Potts Model (#seg=22)
Classical Potts Model (#seg=24)
Classical Potts Model (#seg=55)
Dow Jones 8025.0 ▼ -51.29
Nasdaq 1679.41 ▼ -14.88
FTSE 100 4086.52 ▼ -80.49
Dax 4574.47 ▼ -119.6
Cac 40 3030.25 ▼ -72.18

Fig. 4. Comparison of the extended and classical Potts model for image segmentation.

$$D(y_p, C^t) = \min_h \|y_p - M_h^t\|, \ t \in \{0, 1\}. \tag{9}$$

From (9), we can also define the distance between two components as

$$\text{Dist}(C^0, C^1) = \min_{t \in \{0,1\}} \left(\min_i D(M_i^t, C^{1-t}) \right). \tag{10}$$

Note that in (8), α controls the influence of likelihood energy $E_{\text{lik}}^{\text{b}}$ on the Gibbs energy E^{b}. The larger the α is, the more important is the likelihood energy $E_{\text{lik}}^{\text{b}}$.

The coherence energy measures the segmentation feasibility that represents the contextual prior. We define the coherence energy by extending the classical Potts model to be data-driven [29]:

$$E_{\text{coh}}^{\text{b}}(x_p, x_q) = \frac{|x_p - x_q|}{\|p - q\|} \exp\left(- \frac{\|y_p - y_q\|}{\beta} \right). \tag{11}$$

Our definition of $E_{\text{coh}}^{\text{b}}$ encourages close pixels with similar colors to have the same label. In (11), β controls the influence of the coherence energy $E_{\text{coh}}^{\text{b}}$ on the Gibbs energy E^{b}. The larger the β is, the greater the role that the coherence energy $E_{\text{coh}}^{\text{b}}$ plays in E^{b}.[3] Fig. 4 compares the extended Potts model and the classical Potts model for image segmentation. We can clearly see that the extended model makes the coherence prior more flexible since, with a comparable number of segments, the extended data-driven

[3] It has been shown that the Ising/Potts models cannot provide sufficient smoothing guidance if used as an image-specific prior [48]. As a general (i.e., image-independent) smoothing prior, however, the Potts model has been widely used in MRF labeling [21], [22], [23], [24], [29], [36], [37], [38].

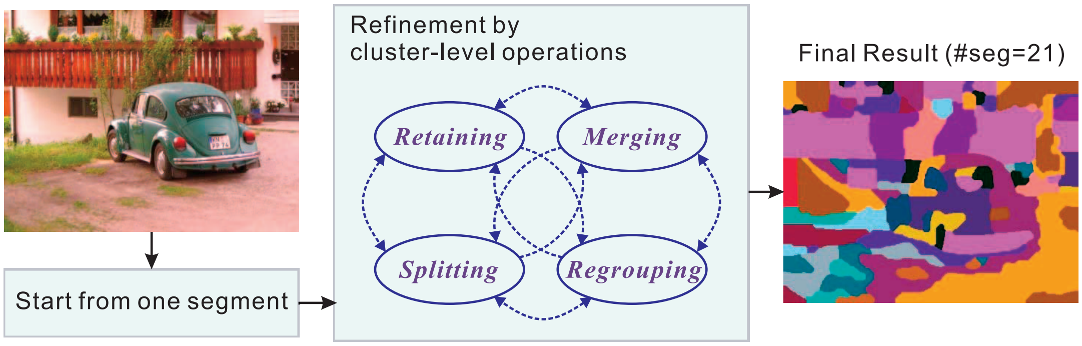


Fig. 5. Working flow of the proposed GGC scheme.

Potts model generates more accurate labeling, while the classical Potts model needs more than twice the number of segments to achieve the similar accuracy.

4 Graduated Graph Cuts

In this section, we will show how to vertically extend the binary s-t graph cut to solve the self-validated labeling problem for image segmentation. The core of our approach is to gradually refine the labeling according to the Gibbs energy minimization principle. Our segmentation refinement is based on the following four types of segment-level operations:

1. *Retaining* $(=)$: For segment S, keeping S unchanged, as denoted by $S = S$.
2. *Splitting* $(\rightarrow)$: For segment S, splitting S into two isolated parts $\tilde{S}^0$ and $\tilde{S}^1$ with minimal binary Gibbs energy (7) using s-t graph cut, as denoted by $S \rightarrow (\tilde{S}^0, \tilde{S}^1)$.
3. *Merging* $(\cup)$: For segment S, merging S with its nearest neighbor $\mathrm{Near}(S)$ to form a new segment S', expressed as $S \cup \mathrm{Near}(S) = S'$. $\mathrm{Near}(S)$ is defined by Definition 3.
4. *Regrouping*: For each segment pair (S_i, S_j), merging them together and then resplitting the merged segment to S'_i and S'_j using binary s-t graph cut to refine the labeling. This process is denoted as $S_i \cup S_j \rightarrow (S'_i, S'_j)$. Segment regrouping is particularly useful in the HGC algorithm to refine the labeling inherited from the coarse scale. A detailed introduction can be found in Section 4.3.

Fig. 5 shows the working flow of the proposed GGC scheme. In general, it is a top-down refinement process. The image is treated as one segment at first. Each segment S is modeled by a binary MRF X^{b}, based on which we iteratively refine the labeling by adjusting each segment with the optimal operation according to the energy minimization principle. This refinement process is terminated when all segments remain unchanged. Since optimizing a binary MRF with s-t graph cut is much easier than solving a flat K-label MRF, the GGC labeling is much faster than classical MRF methods [26]. Using different operations and optimization structures, we propose three segmentation algorithms: tree-structured graph cuts (TSGC), net-structured graph cuts (NSGC), and hierarchical graph cuts (HGC). We will elaborate each algorithm in the following.

4.1 Tree-Structured Graph Cuts (TSGC)

We first analyze the situation of allowing only two types of operations, i.e., segment retaining and splitting. Accordingly, the GGC scheme shown in Fig. 5 is concretized to a tree-structured evolution process, as shown in Fig. 6a.

At the beginning, the image I is treated as one segment, i.e., $\mathrm{Seg}^{(0)}(I) = \{S_0\}$. The initial segment S_0 can be modeled as a binary MRF X^{b}_0. Based on the nonparametric feature model $F(S_0)$ of segment S_0 and corresponding energy

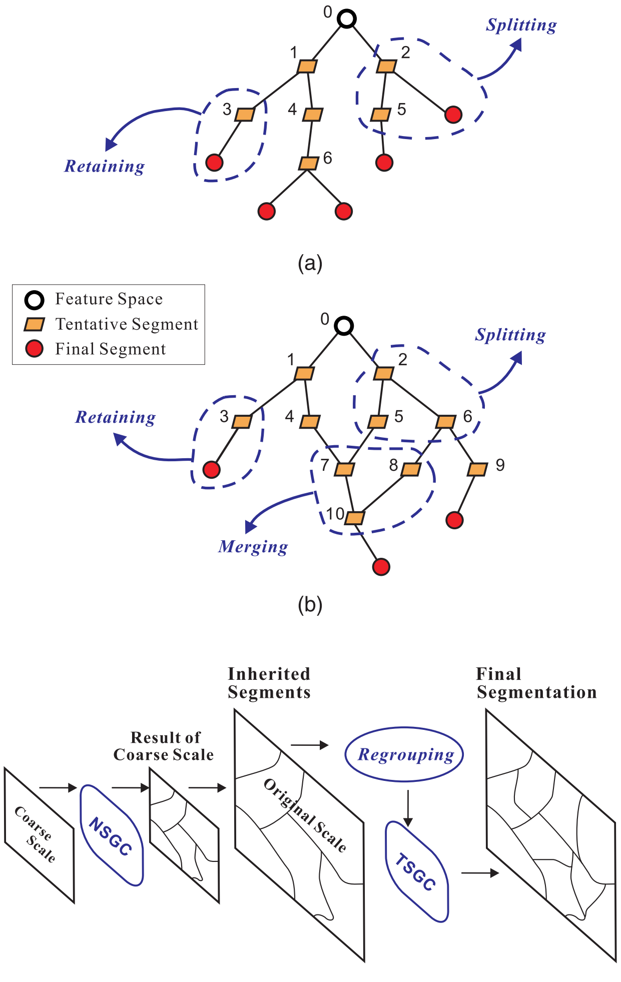


Fig. 6. Algorithmic structure of the proposed methods: (a) TSGC, (b) NSGC, and (c) HGC.

assignment (7)-(11), we can obtain an optimal *splitting* of $S_0 \rightarrow (\tilde{S}_0^0, \tilde{S}_0^1)$ in terms of Gibbs energy minimization, where $\tilde{S}_0^0$ and $\tilde{S}_0^1$ are two tentative segments derived from S_0 using binary s-t graph cut. If the splitting is *better* than retaining S_0 unchanged, we update S_0 to $\tilde{S}_0^0$ and $\tilde{S}_0^1$. Accordingly, the segmentation is updated to $\mathrm{Seg}^{(1)}(I) = \{S_1, S_2\}$ with $S_1 = \tilde{S}_0^0$ and $S_2 = \tilde{S}_0^1$. We then model each segment S_i separately with binary MRF X_i^{b} and find the best operation according to the principle of energy minimization. We continue this process iteratively until, for all segments, operation retaining is better than splitting. We can see that this is an iterative labeling evolution process, which is terminated when all tentative segments remain unchanged; then we obtain the final segmentation $\mathrm{Seg}(I)$. In the following, we give formalized description of the TSGC algorithm:

Definition 1. *Given two operations A and B, we say A is better than B for segment S iff one of the following two conditions satisfied:*

1. $E_{\mathrm{lik}}(A\langle S\rangle) + E_{\mathrm{coh}}(A\langle S\rangle) < E_{\mathrm{lik}}(B\langle S\rangle) + E_{\mathrm{coh}}(B\langle S\rangle)$;
2. $E_{\mathrm{lik}}(A\langle S\rangle) + E_{\mathrm{coh}}(A\langle S\rangle) = E_{\mathrm{lik}}(B\langle S\rangle) + E_{\mathrm{coh}}(B\langle S\rangle)$ *and* $\#\mathrm{seg}(A\langle S\rangle) < \#\mathrm{seg}(B\langle S\rangle)$.

Herein, $A\langle S\rangle$ *and* $B\langle S\rangle$ *represent the tentative segments generated by operation A and B, respectively, and* $\#\mathrm{seg}(\cdot)$ *represents the number of segments.*

Definition 2. *A segment S is* splittable *iff operation splitting is better than retaining for segment S, i.e.,* $E_{\mathrm{split}}(S) < E_{\mathrm{retain}}(S)$ *(see (12) and (13) for detailed definitions).*

4.1.1 Segment Retaining Energy

For segment S, its retaining energy is defined as the normalized likelihood energy, or, in other words, its average intrasegment distance

$$E_{\mathrm{retain}}(S) = \frac{E_{\mathrm{lik}}(S)}{|S|}, \tag{12}$$

where $|S|$ is the cardinality of S.

4.1.2 Segment Splitting Energy

For segment S, we can define its splitting energy as

$$\begin{aligned} E_{\mathrm{split}}(S) &= E_{\mathrm{retain}}(S \rightarrow (\tilde{S}^0, \tilde{S}^1)) \\ &= \frac{E_{\mathrm{lik}}(\tilde{S}^0) + E_{\mathrm{lik}}(\tilde{S}^1) + E_{\mathrm{coh}}(\tilde{S}^0, \tilde{S}^1)}{|S|}. \end{aligned} \tag{13}$$

Refer to (3), (5), (8), and (11) for concrete definitions of $E_{\mathrm{lik}}(S)$ and $E_{\mathrm{coh}}(S_i, S_j)$, respectively.

Specifically, for segment S, the likelihood energy $E_{\mathrm{lik}}(S)$, referring to (8), can be viewed as the overall distance between all feature samples to the subcomponents, i.e., $E_{\mathrm{lik}}(S) = \sum_{y_p \in F(S)} D(y_p, C^{x_p})$. If treating S as one segment, we use $2H$ subcomponents to construct its feature model $F(S)$. By splitting S into two segments $\tilde{S}^0$ and $\tilde{S}^1$, we use $4H$ subcomponents to represent $F(S)$. Since all subcomponents are derived by K-means, it is empirically true that more subcomponents will result in less overall intrasegment distance. Therefore, we have

$$E_{\mathrm{lik}}(\tilde{S}^0) + E_{\mathrm{lik}}(\tilde{S}^1) \leq E_{\mathrm{lik}}(S). \tag{14}$$

Besides, for the Potts-based coherence energy (11), it is easy to see that $E_{\mathrm{coh}}(x_p, x_q) \geq 0$ and $E_{\mathrm{coh}}(x_p, x_q) = 0$ if $x_p = x_q$; thus the following property satisfies:

$$E_{\mathrm{coh}}(S_i, S_j) \geq 0 \quad \text{and} \quad E_{\mathrm{coh}}(S, S) = 0, \tag{15}$$

4.1.3 The Algorithm

Properties (14) and (15) reflect that operation splitting may result in more accurate segmentation than retaining, but operation retaining guarantees the spatial coherence. Hence, the final segmentation of TSGC is a proper balance between the accuracy and spatial coherence based on the principle of Gibbs energy minimization. Algorithm 2 gives the detailed process of TSGC segmentation.

Algorithm 2. TSGC Segmentation

```
Input: Input image $I$.
Output: Final segmentation $\mathrm{Seg}(I)$.
1. Treat $I$ as one segment $S_0$ and Set $\#\mathrm{change} := 1$;
2. Start the TSGC process:
    while $\#\mathrm{change} \neq 0$ do
        Set #change := 0;
        for $i = 1$ to #segments do
            Derive the two-level nonparametric feature
            model $F(S_i)$ by Algorithm 1;
            Compute $E_{\mathrm{retain}}(S_i)$ and $E_{\mathrm{split}}(S_i)$;
            if $S_i$ is splittable then
                $\#\mathrm{change}{+}{+}$;
                Split $S_i$ by s-t graph cut: $S_i \rightarrow (\tilde{S}^0, \tilde{S}^1)$;
            else
                Remain $S_i$ unchanged: $S_i = S_i$;
            end
        end
    end
```

It is easy to prove that in each iteration of TSGC the overall energy decreases. Formally speaking, let $\mathrm{Seg}^{(t)}(I)$ represent the tentative segmentation of image I after t iterations of TSGC. If $\mathrm{Seg}^{(t+1)}(I) \neq \mathrm{Seg}^{(t)}(I)$, then $\mathrm{Seg}^{(t+1)}(I)$ is better than $\mathrm{Seg}^{(t)}(I)$ in that $E(\mathrm{Seg}^{(t+1)}(I)) < E(\mathrm{Seg}^{(t)}(I))$. This means that TSGC will finally converge to a local minimum of the energy function. Its goodness will be further discussed in Section 4.4. Since each segment updates using the optimal operation in each iteration, TSGC guarantees stepwise optimum.[4] TSGC converts the K-class segmentation problem to a series of binary segmentation subproblems. Since it starts from regarding the whole image as one segment and evolves based on the energy minimization principle, it is self-validated. However, as shown in Fig. 7, the rigid tree structure might lead to the overpartitioning problem.

4.2 Net-Structured Graph Cuts (NSGC)

To remedy the overpartitioning problem of TSGC, we introduce another operation, i.e., segment merging. With

[4] We say the proposed algorithms guarantee stepwise optimum, the meaning of which is two-fold: 1) In each iteration, they select the best operation for each tentative segment, thus it is stepwisely optimal in the algorithm configuration; and 2) the binary splitting for each tentative segment is globally optimal due to the s-t graph cut, which locally contributes to the labeling refinement.

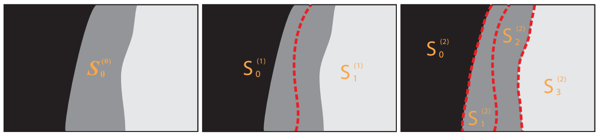


Fig. 7. Illustration of overpartitioning problem of the TSGC algorithm: (a) The synthetic image is first treated as one initial segment $S_0^{(0)}$; (b) after one iteration it splits into two segments $S_0^{(1)}$ and $S_1^{(1)}$; (c) finally, the synthetic image is segmented into four segments $\{S_i^{(2)}\}_{i=0}^{3}$. It is obvious that $S_1^{(2)}$ and $S_2^{(2)}$ are erroneously partitioned, which can be refined by an additional operation, i.e., segment merging.

three types of operations, i.e., segment retaining, splitting, and merging, the GGC scheme becomes a net-structured segmentation process, i.e., NSGC, as shown in Fig. 6b. Similarly to TSGC, NSGC also starts from regarding image I as one segment $\mathrm{Seg}^{(0)}(I) = \{S_0\}$, and iteratively selects the best operation for each segment according to the energy minimization principle. This refinement process is terminated when all tentative segments remain unchanged. Then we obtain the final segmentation $\mathrm{Seg}(I)$. Clearly, NSGC is self-validated.

To quantize the optimality of operation merging, we need a similarity measure of two segments S_i and S_j,

$$\begin{aligned}\mathrm{Dist}(S_i, S_j) &= \mathrm{Dist}(F(S_i), F(S_j))\\ &= \min_{a,b\in\{0,1\}} (\mathrm{Dist}(C^a(S_i), C^b(S_j))),\end{aligned} \tag{16}$$

where $F(S) = \{C^0(S), C^1(S)\}$ represents the two-level nonparametric feature model of segment S. We measure the distance between two segments as their minimal components distance, which is defined in (10).

Definition 3. *In segmentation* $\mathrm{Seg}(I) = \{S_i\}_{i=1}^K$, *for segment* S_i, *we say* S_j *is its* nearest neighbor *iff* $j \neq i$ *and* $S_j = \arg\min_{1\leq j\leq K}\mathrm{Dist}(S_i, S_j)$. *We use* $\mathrm{Near}(S_i)$ *to represent the nearest neighbor of* S_i.

Definition 4. *A segment* S *is* mergeable *to its nearest neighbor* $\mathrm{Near}(S)$ *iff* $E_{\mathrm{merge}}(S, \mathrm{Near}(S)) \leq Q$, *where* Q *is defined in (17). The definition of* E_{retain}, E_{split}, *and* E_{merge} *can be found in (12), (13), and (18).*

$$Q = \frac{1}{|S\cup\mathrm{Near}(S)|} \min\left(\begin{array}{r} E_{\mathrm{retain}}(S)\cdot|S| + E_{\mathrm{retain}}(\mathrm{Near}(S))\cdot|\mathrm{Near}(S)|\\ +\, E_{\mathrm{coh}}(S,\mathrm{Near}(S))\\ E_{\mathrm{split}}(S)\cdot|S| + E_{\mathrm{retain}}(\mathrm{Near}(S))\cdot|\mathrm{Near}(S)|\\ +\, E_{\mathrm{coh}}(S,\mathrm{Near}(S))\\ E_{\mathrm{retain}}(S)\cdot|S| + E_{\mathrm{split}}(\mathrm{Near}(S))\cdot|\mathrm{Near}(S)|\\ +\, E_{\mathrm{coh}}(S,\mathrm{Near}(S))\\ E_{\mathrm{split}}(S)\cdot|S| + E_{\mathrm{split}}(\mathrm{Near}(S))\cdot|\mathrm{Near}(S)|\\ +\, E_{\mathrm{coh}}(S,\mathrm{Near}(S)) \end{array}\right). \tag{17}$$

4.2.1 Segment Merging Energy

For segment S, its merging energy is defined as the retaining energy of the merged segment $S \cup \mathrm{Near}(S)$,

$$\begin{aligned}E_{\mathrm{merge}}(S, \mathrm{Near}(S)) &= E_{\mathrm{retain}}(S\cup\mathrm{Near}(S))\\ &= \frac{E_{\mathrm{lik}}(S\cup\mathrm{Near}(S))}{|S\cup\mathrm{Near}(S)|}.\end{aligned} \tag{18}$$

For segment S, Property (14) leads to the following property of the merging energy (18):

$$E_{\mathrm{merge}}(S, \mathrm{Near}(S)) \geq \frac{E_{\mathrm{lik}}(S) + E_{\mathrm{lik}}(\mathrm{Near}(S))}{|S\cup\mathrm{Near}(S)|}. \tag{19}$$

4.2.2 The Algorithm

Properties (14), (15), and (19) tell us that operation splitting will result in more accurate segmentation, operation merging will lead to more spatially coherent labeling, and operation retaining keeps the good segments unchanged. In NSGC, the final segmentation is a proper evolution result of the three types of operations. The introduction of operation merging will circumvent the overpartitioning problem of TSGC. Algorithm 3 is the detailed process of NSGC segmentation.

Algorithm 3. NSGC Segmentation.

```
Input: Input image $I$.
Output: Final segmentation $\mathrm{Seg}(I)$.
1. Treat $I$ as one segment $S_0$ and Set $\#\mathrm{change} := 1$;
2. Start the NSGC process:
    while $\#\mathrm{change} \neq 0$ do
        Set $\#\mathrm{change} := 0$;
        for $i = 1$ to $\#\mathrm{segments}$ do
            Derive the two-level nonparametric feature
            model $F(S_i)$ using Algorithm 1;
            Derive the nearest neighbor $\mathrm{Near}(S_i)$;
            Compute $E_{\mathrm{retain}}(S_i)$, $E_{\mathrm{split}}(S_i)$ and
            $E_{\mathrm{merge}}(S_i, \mathrm{Near}(S_i))$;
            if $S_i$ is mergeable then
                $\#\mathrm{change}{+}{+}$;
                Merge $S_i$ with $\mathrm{Near}(S_i)$: $S_i \cup \mathrm{Near}(S_i)$;
            else if $S_i$ is splittable then
                $\#\mathrm{change}{+}{+}$;
                Split $S_i$ by s-t graph cut: $S_i \rightarrow (\tilde{S}^0, \tilde{S}^1)$;
            else
                Remain $S_i$ unchanged: $S_i = S_i$;
            end
        end
    end
```

Formally, let $\mathrm{Seg}^{(t)}(I)$ represent the tentative segmentation of image I after t iterations of NSGC. We have $E(\mathrm{Seg}^{(t+1)}(I)) \leq E(\mathrm{Seg}^{(t)}(I))$. Note that if $E(\mathrm{Seg}^{(t+1)}(I)) < E(\mathrm{Seg}^{(t)}(I))$, it is obvious that $\mathrm{Seg}^{(t+1)}(I)$ is better than $\mathrm{Seg}^{(t)}(I)$. If $\mathrm{Seg}^{(t+1)}(I) \neq \mathrm{Seg}^{(t)}(I)$ and $E(\mathrm{Seg}^{(t+1)}(I)) = E(\mathrm{Seg}^{(t)}(I))$, all segments of $\mathrm{Seg}^{(t+1)}(I)$ must be derived from $\mathrm{Seg}^{(t)}(I)$ by either operation retaining or merging. Since $\mathrm{Seg}^{(t+1)}(I) \neq \mathrm{Seg}^{(t)}(I)$, at least one segment in $\mathrm{Seg}^{(t+1)}(I)$ is derived by operation merging. That is, $\mathrm{Seg}^{(t+1)}(I)$ has a smaller number of segments than $\mathrm{Seg}^{(t)}(I)$ and the same overall Gibbs energy. Hence, by Occam's razor, we can still say that $\mathrm{Seg}^{(t+1)}(I)$ is a better segmentation than $\mathrm{Seg}^{(t)}(I)$.

4.3 Hierarchical Graph Cuts (HGC)

Both TSGC and NSGC gradually approach a suboptimal segmentation and guarantee stepwise optima. For each segment S, the major computation burden of TSGC and NSGC lies in three aspects: 1) establishing the nonparametric feature model, 2) optimizing the binary MRF by s-t graph cut, and 3) evaluating operation energies of $E_{\mathrm{retain}}(S)$, $E_{\mathrm{split}}(S)$, and $E_{\mathrm{merge}}(S)$. All of these aspects can be solved in polynomial time; thus, both TSGC and NSGC are of polynomial complexity. Furthermore, we have the following observations:

1. Both TSGC and NSGC can start from any initial labeling.
2. For a graph with n vertices and m arcs, the worst-case complexity of s-t graph cut is $O(mn^2)$ [45], [46]. Recalling the graph formulation of image segmentation, we can find that a smaller image means fewer vertices and much fewer arcs. This implies that we could further improve the efficiency by running TSGC and NSGC on an image pyramid.
3. In a two-level image pyramid, the distance between two segments at the original scale is larger than that at the coarse scale because segments at the original scale may contain much more detail than the coarse scale. Therefore, if a segment S is not mergeable at the coarse scale, it is almost impossible for it to be mergeable at the original scale.

These observations motivate a hierarchical graph cuts (HGC) algorithm, as shown in Fig. 6c. For an image I, we first establish a two-level Gaussian pyramid. Let $\mathrm{Lev}_{\cos}(I)$ be the downsampled image at coarse scale and $\mathrm{Lev}_{\mathrm{org}}(I)$ be the image at original scale. We first use NSGC to obtain an initial labeling of $\mathrm{Lev}_{\cos}(I)$ at the coarse scale. The labeling $\mathrm{Seg}(\mathrm{Lev}_{\cos}(I))$ is then refined by the operation regrouping at the original scale, which leads to an initial segmentation $\mathrm{Seg}'(\mathrm{Lev}_{\cos}(I))$ at the original scale. Finally, starting from $\mathrm{Seg}'(\mathrm{Lev}_{\cos}(I))$, we run TSGC to obtain the final segmentation $\mathrm{Seg}(\mathrm{Lev}_{\mathrm{org}}(I))$. The detailed process of HGC segmentation is given in Algorithm 4. We will show in Section 5 that the HGC algorithm apparently improves the computational efficiency of TSGC and NSGC and produces comparable results.

```
Algorithm 4. HGC Segmentation
  Input: Input image $I$.
  Output: Final segmentation $\mathrm{Seg}(I)$.
  1. Establish a two-level Gaussian pyramid for image
     $I$: $\{\mathrm{Lev}_{\cos}(I), \mathrm{Lev}_{\mathrm{org}}(I)\}$;
  2. Run NSGC algorithm at the coarse scale and obtain
     $\mathrm{Seg}(\mathrm{Lev}_{\cos}(I))$;
  3. Regrouping the inherited segmentation at the original
     scale and obtain $\mathrm{Seg}'(\mathrm{Lev}_{\cos}(I))$;
  4. Using $\mathrm{Seg}'(\mathrm{Lev}_{\cos}(I))$ as initialization, run TSGC
     algorithm to obtain $\mathrm{Seg}(\mathrm{Lev}_{\mathrm{org}}(I))$;
  5. Set $\mathrm{Seg}(I) := \mathrm{Seg}(\mathrm{Lev}_{\mathrm{org}}(I))$;
```

Note that the HGC algorithm can also work on a multilevel pyramid. However, we recommend using a two-level pyramid because of two reasons. First, observation (3) indeed implies a two-level structure for HGC, i.e., considering retaining, splitting, and merging at the coarse scale while considering retaining and splitting only at the original scale. Second, although a multilevel partitioning can be used to balance the accuracy and efficiency [40], the main motive of HGC is for acceleration. As shown by our experiments, we can already obtain an apparent speedup using a two-level pyramid. As aforementioned, one major computation burden of our approach lies in the computation of retaining, splitting, and merging energies. The speedup of HGC mainly comes from $\mathrm{TimeA} + \mathrm{TimeB} - \mathrm{TimeC}$, where TimeA refers to the time reduced by the avoidance of merging energy evaluation at the finer scale, TimeB is the time reduced by running NSGC at the coarse scale, and TimeC is the additional time of refining the labeling inherited from the coarser scale by regrouping operation at the finer scale. Note that TimeC will be longer if there are more segments needed to be refined. For a multilevel pyramid, the accumulated splitting operations at several finer scales might result in many tentative segments, thus increasing TimeC and depressing the time reduction. Although, using multilevel pyramid, we may still achieve time reduction with carefully tuned parameters at each scale, according to our tests, we find that a two-level pyramid performs well in acceleration and needs much less effort in parameter tuning.

4.4 Suboptimality Discussion

We now empirically validate the suboptimality of the proposed algorithms based on a general criterion, and qualitatively explain the rationale behind the observations. Note that the only difference between the self-validated labeling and K-labeling segmentation is that the number of segments K is unknown in our formulation. Except for it, the energy function, as defined by (1), (7), (8), and (11), of our scheme also applies to K-labeling segmentation. Therefore, we can measure the goodness of our labeling results in terms of the standard K-labeling energy function by using the particular K generated by our algorithms.

We use the pixel labels changing (PLC) ratio to measure the labeling consistency of two segmentations,

$$\mathrm{PLC}(\mathrm{Seg}_1,\ \mathrm{Seg}_2) = \frac{1}{n}\ \mathrm{Inc}(\mathrm{Seg}_1, \mathrm{Seg}_2), \tag{20}$$

where $\mathrm{Inc}(\mathrm{Seg}_1, \mathrm{Seg}_2)$ indicates the number of pixels with inconsistent labels in Seg_1 and Seg_2, n is the number of pixels in the image. For a given image, we first run our algorithms and then refine the labeling using α-β-swap and α-expansion, two state-of-the-art techniques to reach bounded local minima for K-labeling problems [21]. The PLC ratio between our labeling and the refined one represents the closeness of our results to the bounded local minima, thus objectively reflecting the goodness of our solutions.

To derive an unbiased validation, we constructed 200 test cases that include 40 images and tried 5 different sets of parameters for each image. Table 1 shows the results, from which we have two major observations. First, the segmentations of our algorithms are quite close to the local minima obtained by α-β-swap and α-expansion (using our labeling as initialization and using the same K determined by our algorithms). For the 200 test cases, the average PLC ratio of TSGC, NSGC and HGC is lower than 12 percent for α-expansion and lower than 5 percent for α-β-swap. Second, although only less than 12 percent pixels may change their labels, for a particular K, the refinement process is about 3 to 6 times slower than the proposed algorithms to obtain a self-validated labeling. Thus, it is highly infeasible by trying multiple K using α-β-swap or α-expansion to seek the best number of labels, especially for segmenting a large number of images. Fig. 8 shows an example of NSGC segmentation and the refinement results. It is clear that the segmentations are visually quite similar and only a very small part of labels are

TABLE 1
Average PLC Ratio (PR) and Running-time Ratio (TR, i.e., RefineTime/AlgTime) of Our Algorithms for 200 Test Cases

Method		TSGC	NSGC	HGC
α-β -swap	PR	$5.0\% \pm 3.8\%$	$4.3\% \pm 3.0\%$	$4.9\% \pm 5.9\%$
	TR	4.9 ± 0.9	3.8 ± 0.8	6.5 ± 1.6
α-exp	PR	$11.8\% \pm 3.9\%$	$10.9\% \pm 3.9\%$	$11.5\% \pm 5.4\%$
	TR	4.5 ± 0.7	3.5 ± 0.6	6.0 ± 1.1

changed. It is also interesting to note that the PLC ratio of TS-MRF by α-expansion is only 0.34 percent, but clearly the segmentation is not good due to lack of spatial coherence. Thus, Fig. 8 also reflects that α-β-swap and α-expansion (i.e., the horizontal extensions to binary s-t graph cut) are very sensitive to the initialization. Consequently, we have empirically validated that our algorithms can obtain good local minima with much less time than trying multiple K by α-β-swap and α-expansion (see the time ratio reported in Table 1).

Note that the above suboptimality discussion is derived from empirical experiments. Unlike stereo and image restoration [21], in image segmentation, the feature model of each segment needs to be iteratively updated. Thus, it is difficult to strictly analyze the energy bound using the similar way of α-expansion [21]. We can only qualitatively analyze the two major reasons making our self-validated labelings cannot be largely refined by α-β-swap. First, for those segment pairs generated by a splitting operation of the same tentative segment, the α-β-swap refinement is equivalent to repeating the splitting using twice number of subcomponents. Hence, with proper subcomponent number H, the globally optimal binary splitting guarantees that almost all labels of these segment pairs cannot be changed by α-β-swap. Second, referring to the algorithmic structure shown in Fig. 6, any segment pair that is not directly generated by a splitting operation belongs to a subset of some ancestor tentative segment. Again, the globally optimal splitting of the ancestor tentative segment makes most labels of these segment pairs survive under the α-β-swap refinement.

5 Experimental Results

We evaluated the performance of the proposed algorithms for natural image segmentation, and compared them with eight state-of-the-art methods:[5]

1. Efficient graph-based segmentation (GBS) [13],
2. MeanShift [14],
3. Two K-way graph cuts methods: KGC-I (K-means + α-expansion) and KGC-II (K-means + α-β-swap) [21],
4. Isoperimetric graph partitioning (IsoCut) [18],
5. Tree-structured MRF (TS-MRF) [15],
6. Data-driven Markov Chain Monte Carlo (DDMCMC) [16], and
7. Normalized cut (NCut) [6], [33].

Our evaluation was based on three benchmark data sets: the Berkeley database [49], the CityU data set, and the Weizmann database [50]. To maintain a fair comparison, all reported results of our algorithms were automatically obtained using the same parameter setting: $\alpha = 1.2$, $\beta = 0.8$, and $H = 3$.[6] All tested methods used the color information only (in CIE $L^*u^*v^*$ color space).

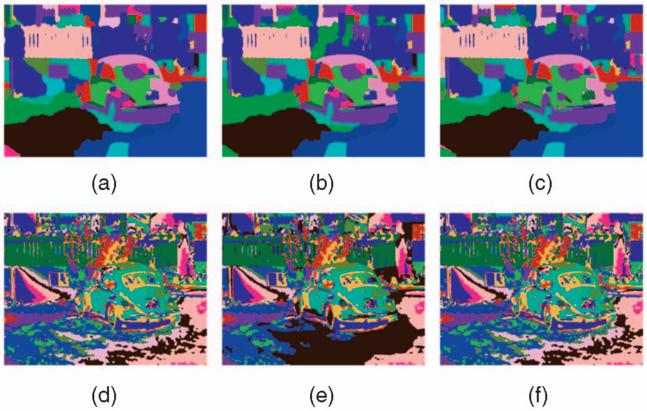


Fig. 8. An example of segmentation refinement by α-β-swap and α-expansion: (a) is the NSGC segmentation result of the car image shown in Fig. 5 with 20 percent noise; (b) is the refined NSGC segmentation by α-β-swap of which only 6.71 percent pixel labels are refined and 93.29 percent pixel labels remain unchanged; (c) shows the refined NSGC segmentation by α-expansion where only 7.51 percent pixel labels are changed; (d), (e), and (f) are the segmentation result of TS-MRF and its refined labeling by α-β-swap and α-expansion, respectively. Note that NSGC and TS-MRF shared the same energy function and parameters. (a) NSGC ($\#\text{seg} = 20$). (b) α-β-swap refinement (6.71 percent refined). (c) α-expansion refinement a (7.51 percent refined). (d) TS-MRF ($\#\text{seg} = 16$). (e) α-β-swap refinement (17.80 percent refined). (f) α-expansion refinement (0.34 percent refined).

We first compared the segmentation accuracy and the labeling cost. As we know, image segmentation partitions an image into several salient regions according to the similarity of low-level features. Since low-level distance cannot always be consistent with the semantic dissimilarity, objective evaluation of image segmentation is still a challenging problem. In practice, segmentation accuracy is usually evaluated either in the context of high-level applications [4] or by comparing segmentations with human-labeled ground truth [49], [50].

Fig. 9 shows the comparative results of human-labeled segmentations and the results of 11 segmentation methods on the Berkeley database. To make an apparent comparison, all segmentations are shown by their segment boundary maps. Among all tested methods, GBS and MeanShift efficiently generate segments according to local similarity or nonlinear filtering of neighboring pixels. This bottom-up structure makes GBS and MeanShift tend to oversegment the image and results in many spurious segments. In contrast, KGC (I and II), IsoCut, TS-MRF, DDMCMC, NCut, and the proposed algorithms have general objective functions and converge to strong local optima, thus leading to a fine balance between labeling accuracy and spatial coherence. See the first image in Fig. 9, for example, GBS and

[5] All tested methods, except for KGC and TS-MRF, used the original implementations of their own authors. KGC (I and II) and TS-MRF used the same energy configuration of the proposed algorithms and different optimization methods.

[6] Note that, for other comparison methods, we used slightly different parameters to segment different images based on a default setting. For each method, the default parameter setting was chosen as the best one by trying several sets of reasonable parameters for 20 validation images. The following lists the default parameters used for each comparison method in our experiments: 1) GBS ($\sigma = 0.45$, $k = 320$, $min = 80$); 2) MeanShift: ($spatial = 7$, $color = 6.5$, $min\_reg = 50$); 3) KGC (K: adjusted for each image, $\alpha = 1.2$, $\beta = 0.8$, $H = 3$); 4) IsoCut ($stop = 0.0004$, $scale = 88$); 5) TS-MRF ($\alpha = 1.2$, $\beta = 0.8$, $H = 3$); 6) DDMCMC ($scale = 3.0$, $stop\_temp = 58.1$); 7) NCut (K: adjusted for each image).

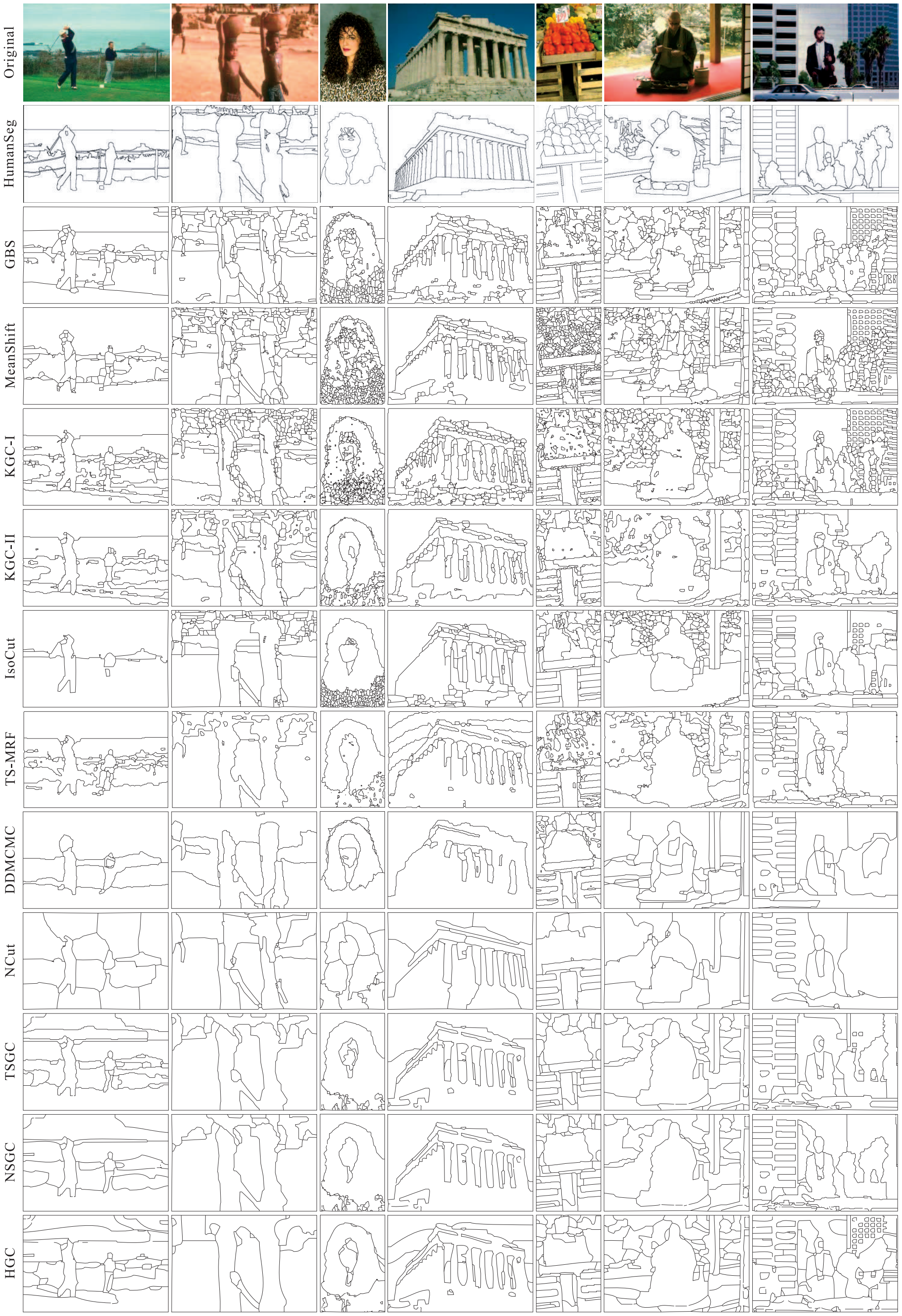
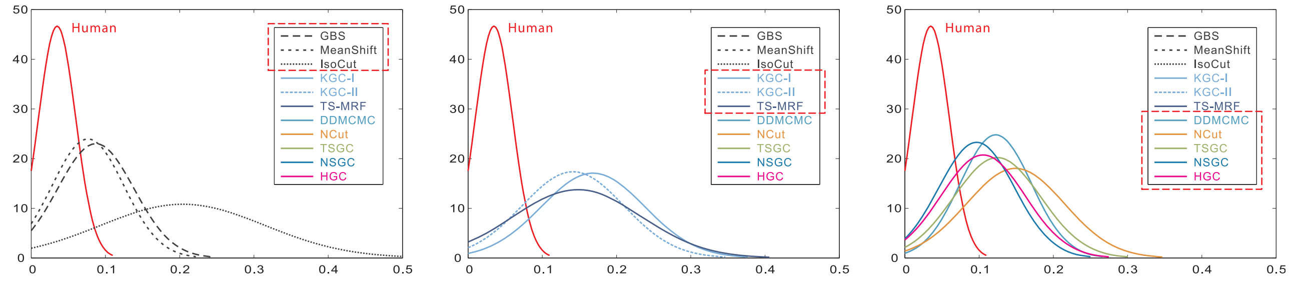


Fig. 9. Comparison of the segmentation accuracy of GBS, MeanShift, KGC (I and II), IsoCut, TS-MRF, DDMCMC, NCut, and the proposed TSGC, NSGC, and HGC algorithms on the Berkeley database.

MeanShift produces 97 and 110 segments, respectively, while TSGC generates 16 segments, NSGC 13 segments, and HGC 18 segments. In all 11 tested methods, GBS, MeanShift, IsoCut, TS-MRF, DDMCMC, and the proposed TSGC, NSGC, and HGC algorithms are self-validated. We can see that, compared to other self-validated methods, our algorithms produce consistent segmentations that are qualitatively similar to the human-labeled ground truth.

Besides, to make an overall quantitative comparison, we used the local consistency error (LCE) [49] to measure the segmentation difference. The LCE measure is defined as the average refinement rate between Seg_1 and Seg_2, which punishes conflict segmentations and allows mutual refinement. Fig. 10 plots the histograms of LCE measure for all tested methods versus human-labeled ground truth on the Berkeley database [49]. In terms of LCE measure, our



(a)

(b)

(c)

Fig. 10. LCE histograms of different methods versus human-labeled ground truth on the Berkeley database.

algorithms are appreciably better than KGC (I and II), IsoCut, TS-MRF, and NCut, and comparable to DDMCMC, GBS, and MeanShift.[7]

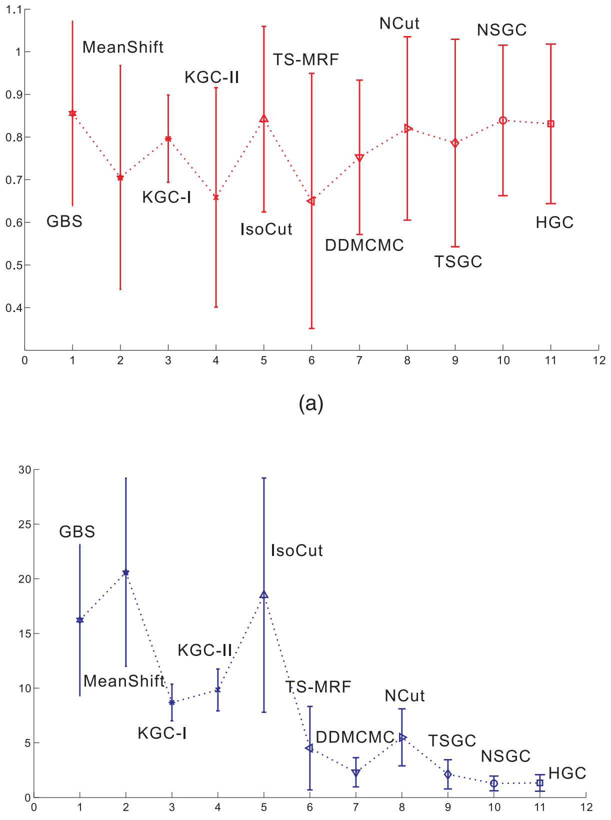
In Fig. 11, we compare the labeling cost of different methods under a comparable accuracy level using the Weizmann database. Fig. 11a shows the average F-measure scores (reflecting the labeling consistency to human-labeled ground truth) and corresponding standard derivations of different methods. As shown in Fig. 11a, all tested methods were tuned to generate comparable labeling accuracy. The labeling cost, i.e., the number of segments to cover an object, is shown in Fig. 11b. We can clearly see that the proposed algorithms need much fewer labels and TS-MRF, DDMCMC, and NCut also performed well.

Second, we tested the segmentation robustness to noise distortion using the CityU data set. The noisy images were synthesized by $I_{\text{noise}} = I_{\text{neat}} + r\,N(0,\,1)$, where I_{neat} and I_{noise} are the neat and noisy image, respectively, $N(0, 1)$ is standard Gaussian noise, and r $(0 \leq r \leq 1)$ controls the noise level. Fig. 12 compares the labeling consistency of 11 methods at increasing noise levels using two measurements, i.e., the LCE measure [49] and the PLC ratio (20). It is clear that NCut, DDMCMC, and the proposed algorithms are quite stable and consistent at different noise levels, while IsoCut is the most unstable one. Fig. 13 shows more results at several noise levels. We found that the bottom-up methods GBS and MeanShift are also very sensitive to noise distortion. At middle noise levels, they tend to generate many meaningless spurious segments and fail to capture the global structure. They have relatively low LCE value in Fig. 12a mainly because of the bias of LCE measurement [49]. As shown in Fig. 12b, the bias of LCE can be rectified by the PLC ratio. From Fig. 13, we can also see that NCut, DDMCMC, and the proposed algorithms outperform the rest methods in terms of robustness to noise, and that KGC-II and TS-MRF are also not very sensitive to noise.

Figs. 9, 10, 11, 12, 13 also demonstrate the general consistency of the proposed algorithms. In fact, TSGC may obtain a comparable labeling quality with a few more labels than NSGC and the introduction of operation merging enables NSGC to produce more reasonable labeling than TSGC, while HGC, working at two scales, may generate more detailed segmentation and is apparently faster than TSGC and NSGC. Table 2 lists the average running time of 11 methods, of which the bottom-up approaches GBS and MeanShift are the fastest ones, and DDMCMC is much slower than other methods.

We show the capability of our algorithms to preserve long-range soft boundaries in Fig. 14. Note that all results of our algorithms reported here were obtained using the simple 4-connected lattice formulation only, referring to Fig. 2a. The desirable capability of preserving soft boundaries of our algorithms mainly results from two aspects: 1) the stepwise optimum property with large move and 2) a proper combination of a generative model (i.e., the likelihood energy for a given tentative segment) and a discriminative model (i.e., the extended data-driven Potts model).

As shown in Fig. 15, coarse-to-fine segmentation is another nice ability of the proposed algorithms. We can clearly see the

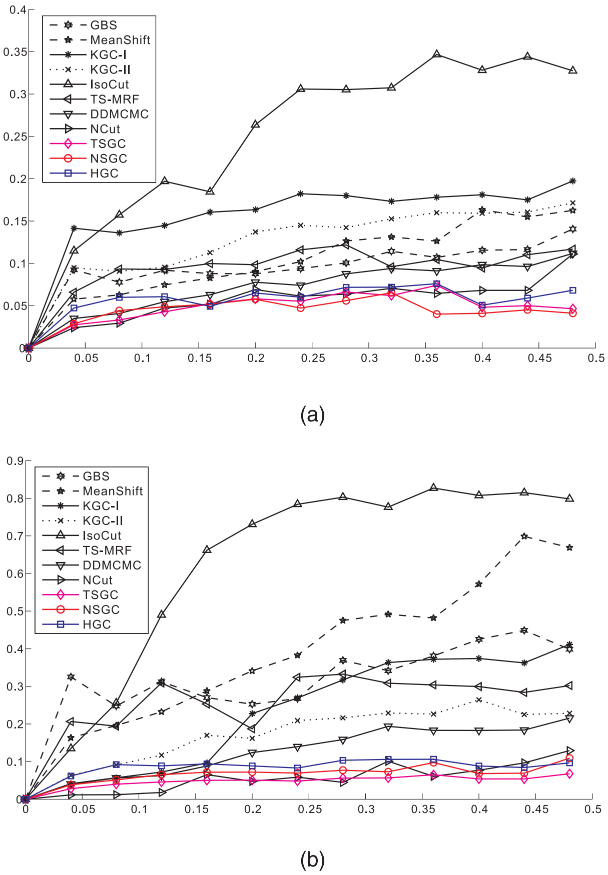


(a)

(b)

Fig. 11. Comparison of segmentation accuracy and labeling cost on the Weizmann database: (a) shows the average F-measure scores and standard deviations of the segmentations to ground truth; (b) shows the average numbers of segments generated to cover an object and standard deviations.

[7] Note that LCE measures the labeling accuracy only and prefers oversegmentation to some extent.



(a)

(b)

Fig. 12. Comparison of labeling consistency at increasing noise levels measured by (a) LCE and (b) PLC ratio. At each noise level, the LCE and PLC values were obtained by averaging the results of 30 images. For a given noise level r, the LCE and PLC values of a method were computed by comparing the segmentations of the noisy image and the neat image.

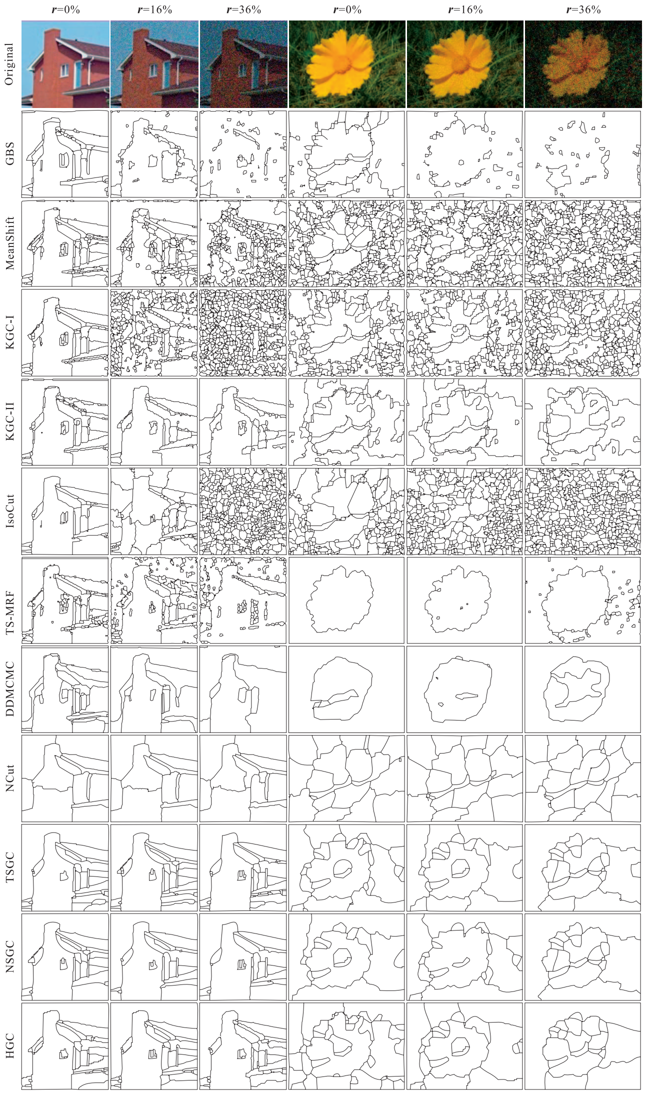


Fig. 13. Robustness to noise distortion. For each test image, the first column shows the noise-free image and corresponding segmentations of different methods, the second column shows the segmentations at noise level 16 percent, and the third column shows the segmentations at noise level 36 percent.

labeling refinement process: Useful details are gradually captured and the global profiles are either reconstructed or preserved. Our algorithms iteratively add, erase, or adjust the tentative segments until the overall Gibbs energy cannot be decreased any more. As a result, the final segmentation reaches a proper balance of the accuracy, spatial coherence, and the labeling cost. We also believe that this hierarchical coarse-to-fine labeling is more useful than a single flat segmentation to reveal the structure of images [19].

In Table 3, we make a general comparison of the 11 segmentation methods from seven aspects: algorithmic structure, spatial coherence, self-validation, robustness to noise, preservation of soft boundaries, oversegmentation, and computational efficiency. Particularly, algorithmic structure refers to whether the segmentation algorithm explores the space of all possible labelings directly or in a hierarchical manner. Segmentation methods usually have three kinds of algorithmic structures: flat (e.g., NCut, KGC-I, and KGC-II), bottom-up (e.g., GBS and MeanShift), and top-down (IsoCut and the proposed algorithms). The spatial coherence is measured from two aspects: 1) the number of isolated fragments in the labeling to obtain a satisfactory accuracy and 2) the consistency between segment boundaries to the object boundaries. As for the efficiency, for images with size 240×160, we define four levels of efficiency: very fast ($\leq$1 s), fast ($\leq$10 s), medium ($\leq$30 s), and slow ($\geq$300 s). The ranking in Table 3 was determined by the overall performance of different methods on three

TABLE 2
Average Running Time of Different Methods for 135 Images Randomly Selected from the Berkeley Data Set (of Size 240×160)

Method	GBS	MeanShift	KGC	IsoCut	TS-MRF	DDMCMC	NCut	TSGC	NSGC	HGC
Time	$<$ 1s	$\approx$ 1s	I: 11.8s II: 15.4s	8.4s	6.5s	652.6s	28.6s	7.0s	9.8s	3.2s

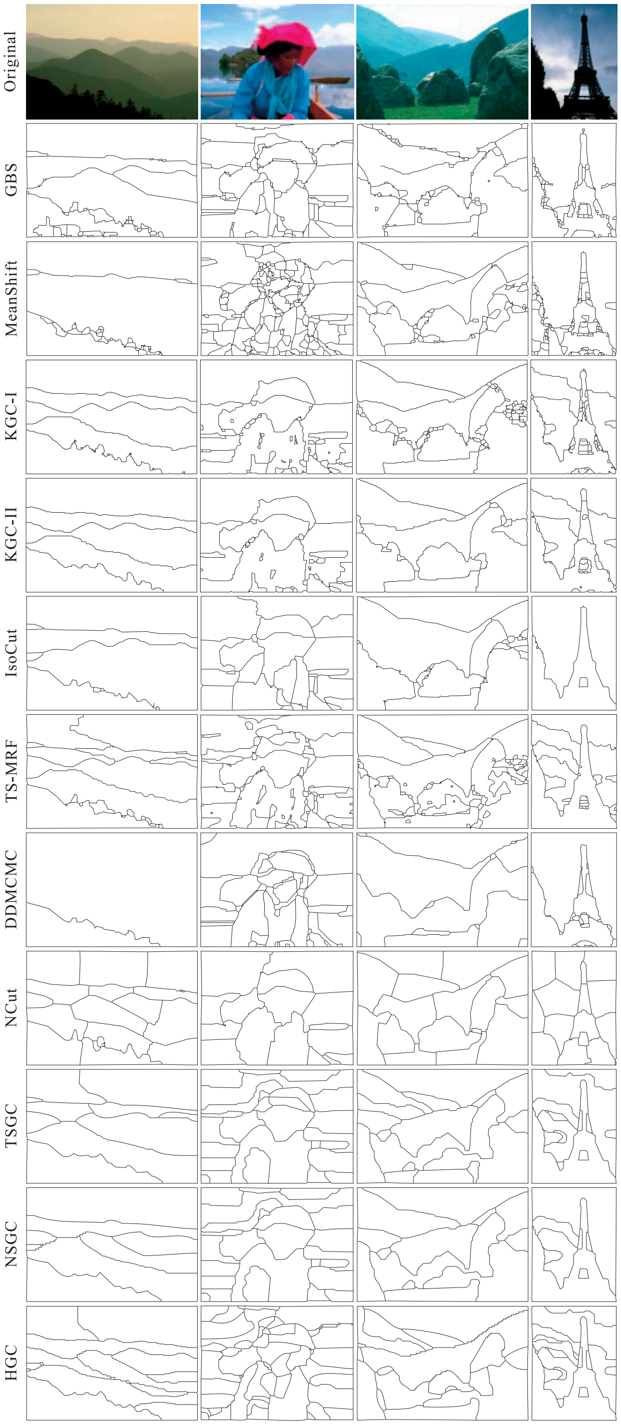


Fig. 14. Preservation of long-range soft boundaries.

benchmark data sets. Table 3 shows that our algorithms have generally satisfying performance in all aspects.

Finally, we compare seven less sensitive methods (TS-MRF, KGC-II, DDMCMC, NCut, and the proposed GGC algorithms) for segmenting two specific images "shape" and "woman" in Fig. 16. Note that TS-MRF and KGC-II are more vulnerable to noise distortion than other methods. DDMCMC is better than TS-MRF and KGC-II, but needs much longer time to converge than other methods. NCut is stable in performance and robust to noise distortion, but it can only return all right segment boundaries only by setting $K = 16$, which is much larger than the "correct" number of segments, 4, for image "shape." And in such a condition, NCut usually returns many oversegmented boundaries besides the right ones. In contrast, the proposed GGC algorithms are able to efficiently produce satisfying results and generate reasonable numbers of segments. Specifically, TSGC tends to generate more segments than NSGC and thus is more suitable to those scenarios where labeling accuracy is more important than labeling cost. HGC hierarchically combines TSGC and NSGC and is faster than TSGC and NSGC with comparable labeling quality and labeling cost. In Fig. 16, we also show the results of a well-known multilevel graph partitioning package METIS [40]. For fairness of comparison, we used the graph construction of NCut in METIS. It is clear that the "pMetis" algorithm performs consistently better than the "kMetis" algorithm, but, as the number of segments increases, both "pMetis" and "kMetis" cannot satisfactorily maintain the spatial coherence.

6 Conclusion and Future Work

In this paper, we have presented a general scheme GGC for self-validated labeling of MRFs and applied to image segmentation. Based on different optimization structures, we have proposed three algorithms: TSGC, NSGC, and HGC, which gradually reconstruct the suboptimal labeling according to the principle of Gibbs energy minimization, and are able to converge to good local minima. Experiments show that the proposed algorithms have generally satisfying properties for image segmentation, and outperform existing methods in terms of robustness to noise, efficiency and preservation of soft boundaries.

Although we only demonstrate the application of our methods for image segmentation, we believe that they are readily applicable to other perceptual grouping and labeling problems in low-level vision, e.g., video segmentation, automatic story segmentation, and multicamera scene reconstruction. Specifically, interesting future work includes:

1. Combining bottom-up methods within the GGC scheme to further improve both the feature model and the efficiency,
2. Studying how to integrate various cues that is the merit of DDMCMC [16],
3. Studying the application of higher-order (image-specific or general) priors in image segmentation [51], [52], and
4. Combining discrete and continuous optimization to solve generic MRFs.

Acknowledgments

The authors would like to thank the associate editor and all of the reviewers for their constructive comments. This work was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China, under General Research Fund (Project Nos.: CUHK 412206, CityU 117806, and 9041369).

TABLE 3
General Comparison of Different Segmentation Methods

Method	Structure	Spatial Coherence	Self Validation	Robustness to Noise	Keep Soft Boundaries	Over Segmentation	Efficiency
GBS	bottom-up	weak	yes	low	no	yes	very fast
MeanShift	bottom-up	weak	yes	low	no	yes	very fast
KGC-I	flat	medium	no	low	yes	no	medium
KGC-II	flat	medium	no	medium	yes	no	medium
IsoCut	top-down	medium	yes	low	no	partial	fast
TS-MRF	top-down	medium	yes	medium	yes	partial	fast
DDMCMC	flat	medium	yes	high	no	no	slow
NCut	flat	strong	no	high	no	no	medium
TSGC	top-down	strong	yes	high	yes	no	fast
NSGC	top-down	strong	yes	high	yes	no	fast
HGC	top-down	strong	yes	high	yes	no	fast

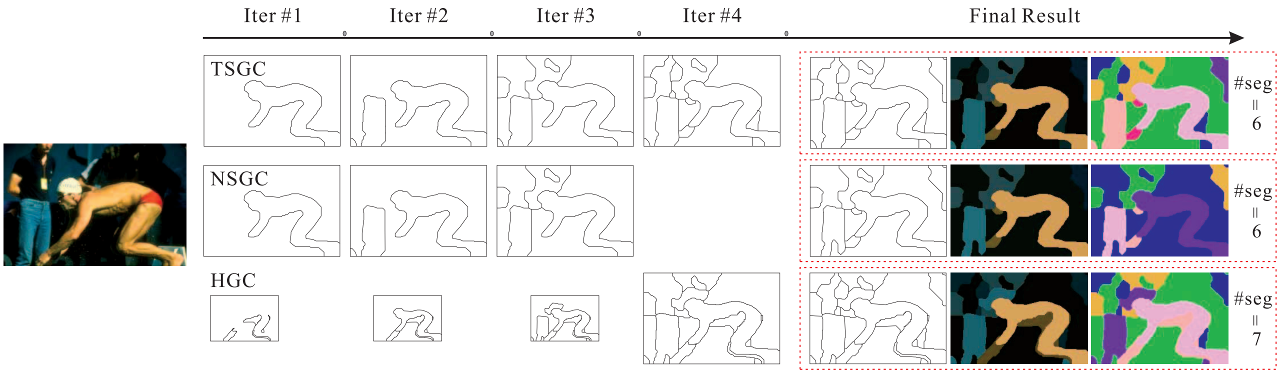


Fig. 15. Hierarchical coarse-to-fine segmentation of the proposed algorithms. For the test image, the proposed methods converge in at most five iterations. In the red-dashed box, the final segmentation results, including boundary map, regional mean color, segment map, and the generated number of segments, are shown.

REFERENCES

[1] S.X. Yu, R. Gross, and J. Shi, "Concurrent Object Segmentation and Recognition with Graph Partitioning," *Proc. Conf. Neural Information Processing Systems,* 2002.

[2] Z. Tu, X. Chen, A. Yuille, and S.-C. Zhu, "Image Parsing: Unifying Segmentation, Detection, and Object Recognition," *Int'l J. Computer Vision,* vol. 63, no. 2, pp. 113-140, 2005.

[3] C. Liu, J. Yuen, and A. Torralba, "Nonparametric Scene Parsing: Label Transfer via Dense Scene Alignment," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[4] W. Feng and Z.-Q. Liu, "Region-Level Image Authentication Using Bayesian Structural Content Abstraction," *IEEE Trans. Image Processing,* vol. 17, no. 12, pp. 2413-2424, Dec. 2008.

[5] Z. Wu and L. Richard, "An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 11, pp. 1101-1113, Nov. 1993.

[6] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 888-905, Aug. 2000.

[7] A.K. Sinop and L. Grady, "A Seeded Image Segmentation Framework Unifying Graph Cuts and Random Walker Which Yields a New Algorithm," *Proc. IEEE 11th Int'l Conf. Computer Vision,* 2007.

[8] D. Singaraju, L. Grady, and R. Vidal, "P-Brush: Continous Valued MRFs with Normed Pairwise Distributions for Image Segmentation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[9] L. Grady, "Random Walks for Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 11, pp. 1768-1783, Nov. 2006.

[10] P. Kohli, A. Shekhovtsov, C. Rother, V. Kolmogorov, and P. Torr, "On Partial Optimality in Multi-Label MRFs," *Proc. Ann. Int'l Conf. Machine Learning,* 2008.

[11] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer, "Generic Object Recognition with Boosting," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 3, pp. 416-431, Mar. 2006.

[12] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On Clustering Validation Techniques," *J. Intelligent Information Systems,* vol. 17, nos. 2-3, pp. 107-145, Dec. 2001.

[13] P.F. Felzenszwalb and D.P. Huttenlocher, "Efficient Graph-Based Image Segmentation," *Int'l J. Computer Vision,* vol. 59, no. 2, pp. 167-181, 2004.

[14] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 5, pp. 1-18, May 2002.

[15] C. D'Elia, G. Poggi, and G. Scarpa, "A Tree-Structured Markov Random Field Model for Bayesian Image Segmentation," *IEEE Trans. Image Processing,* vol. 12, no. 10, pp. 1250-1264, Oct. 2003.

[16] Z. Tu and S.-C. Zhu, "Image Segmentation by Data-Driven Markov Chain Monte Carlo," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 5, pp. 657-673, May 2002.

[17] "Segmentation by Split-and-Merge Techniques," http://iris.usc.edu/Vision-Notes/bibliography/segment347.html, 2010.

[18] L. Grady and E. Schwartz, "Isoperimetric Graph Partitioning for Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 3, pp. 469-475, Mar. 2006.

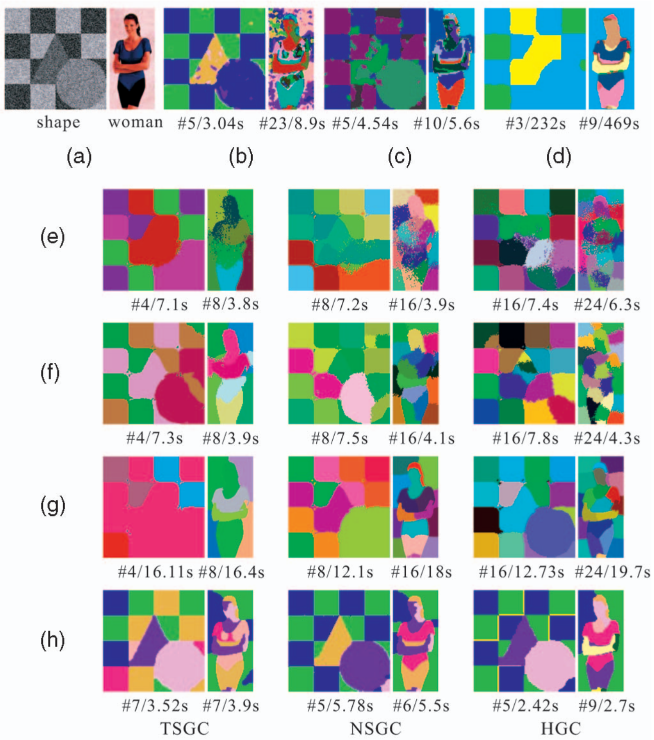


Fig. 16. Comparison of several "robust" segmentation methods and the multilevel graph partitioning package METIS (version 4.0) for synthesized image "shape" and natural image "woman." For each segmentation result, we show the number of segments and the running time in the form of "#seg/time." (a) Original. (b) TS-MRF. (c) KGC-II. (d) DDMCMC. (e) kMetis. (f) pMetis. (g) NCut. (h) GGC.

[19] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt, "Hierarchy and Adaptivity in Segmenting Visual Scenes," *Nature,* vol. 442, no. 7104, pp. 719-846, 2006.

[20] D. Greig, B. Porteous, and A. Seheult, "Exact Maximus A Posteriori Estimation for Binary Images," *J. Royal Statistical Soc., Series B,* vol. 51, no. 2, pp. 271-279, 1989.

[21] Y. Boykov, O. Veksler, and R. Zabih, "Fast Approximate Energy Minimization via Graph Cuts," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 11, pp. 1222-1239, Nov. 2001.

[22] Y. Boykov and V. Kolmogorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 9, pp. 1124-1137, Sept. 2004.

[23] Y. Boykov and G. Funka-Lea, "Graph Cuts and Efficient N-D Image Segmentation," *Int'l J. Computer Vision,* vol. 70, no. 2, pp. 109-131, 2006.

[24] V. Kolmogorov and R. Zabih, "What Energy Functions Can Be Minimized via Graph Cuts?" *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 2, pp. 147-159, Feb. 2004.

[25] V. Kolmogorov and C. Rother, "Minimizing Nonsubmodular Functions with Graph Cuts—A Review," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 7, pp. 1274-1279, July 2007.

[26] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 6, no. 6, pp. 721-741, Nov. 1984.

[27] D. Melas and S. Wilson, "Double Markov Random Fields and Bayesian Image Segmentation," *IEEE Trans. Signal Processing,* vol. 50, no. 2, pp. 357-365, Feb. 2002.

[28] J. Marroquin, E. Santana, and S. Botello, "Hidden Markov Measure Field Models for Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 11, pp. 1380-1387, Nov. 2003.

[29] W. Feng and Z.-Q. Liu, "Self-Validated and Spatially Coherent Clustering with Net-Structured MRF and Graph Cuts," *Proc. Int'l Conf' Pattern Recognition,* vol. 4, pp. 37-40, 2006.

[30] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and Texture Analysis for Image Segmentation," *Intl'l J. Computer Vision,* vol. 43, no. 1, pp. 7-27, 2001.

[31] S.X. Yu and J. Shi, "Segmentation Given Partial Grouping Constraints," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 2, pp. 173-183, Jan. 2004.

[32] S.X. Yu, "Segmentation Using Multiscale Cues," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2004.

[33] T. Cour, F. Bénézit, and J. Shi, "Spectral Segmentation with Multiscale Graph Decomposition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 1124-1131, 2005.

[34] J. Wang and M.F. Cohen, "An Iterative Optimization Approach for Unified Image Segmentation and Matting," *Proc. IEEE Int'l Conf. Computer Vision,* 2005.

[35] J. Wang, M. Agrawala, and M.F. Cohen, "Soft Scissors: An Interactive Tool for Realtime High Quality Matting," *Proc. ACM SIGGRAPH,* 2007.

[36] Y. Boykov and M.-P. Jolly, "Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 105-112, 2001.

[37] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, "Lazy Snapping," *Proc. ACM SIGGRAPH,* pp. 303-308, 2004.

[38] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut—Interactive Foreground Extraction Using Iterated Graph Cuts," *Proc. ACM SIGGRAPH,* pp. 309-314, 2004.

[39] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer, "Optimizing Binary MRFs Via Extended Roof Duality," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.

[40] "METIS Package," http://glaros.dtc.umn.edu/gkhome/views/metis/, 2010.

[41] P. Cheeseman and J. Stutz, "Bayesian Classification (AutoClass): Theory and Results," *Advances in Knowledge Discovery and Data Mining,* pp. 153-180, Am. Assoc. for Artificial Intelligence, 1996.

[42] Y. Li and D. Huttenlocher, "Sparse Long-Range Random Field and Its Application to Image Denoising," *Proc. European Conf. on Computer Vision,* 2008.

[43] A. Elmoataz, O. Lezoray, and S. Bougleux, "Nonlocal Discrete Regularization on Weighted Graphs: A Framework for Image and Manifold Processing," *IEEE Trans. Image Processing,* vol. 17, no. 7, pp. 1047-1060, July 2008.

[44] U. Braga-Neto and J. Goutsias, "Object-Based Image Analysis Using Multiscale Connectivity," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 6, pp. 892-907, June 2005.

[45] E. Dahlhaus, D. Johnson, C. Papadimitriou, P. Seymour, and M. Yannakakis, "The Complexity of Multiway Cuts," *ACM Symp. Theory of Computing,* pp. 241-251, 1992.

[46] J. Evans and E. Minieka, *Optimization Algorithms for Networks and Graphs,* second ed. Marcel Dekker, 1992.

[47] C. Elkan, "Using the Triangle Inequality to Accelerate K-Means," *Proc. Ann. Int'l Conf. Machine Learning,* 2003.

[48] R. Morris, X. Descombes, and J. Zerubia, "The Ising/Potts Model Is Not Well Suited to Segmentation Tasks," *Proc. Digital Signal Processing Workshop,* pp. 263-266, 1996.

[49] D. Martin, C. Fowlkes, and J. Malik, "A Database of Human Segmented Natural Images and Its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics," *Proc. IEEE Int'l Conf. Computer Vision,* vol. 2, pp. 416-423, 2001.

[50] S. Alpert, M. Galun, R. Basri, and A. Brandt, "Image Segmentation by Probabilistic Bottom-Up Aggregation and Cue Integration," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.

[51] C. Rother, P. Kohli, W. Feng, and J. Jia, "Minimizing Sparse Higher Order Energy Functions of Discrete Variables," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[52] P. Kohli, L. Ladicky, and P. Torr, "Robust Higher Order Potentials for Enforcing Label Consistency," *Int'l J. Computer Vision,* vol. 82, pp. 302-324, 2009.

Wei Feng received the BS and MPhil degrees in computer science from Northwestern Polytechnical University, China, in 2000 and 2003, respectively, and the PhD degree in computer science from the City University of Hong Kong in 2008. He worked as a postdoctoral fellow at the Chinese University of Hong Kong. He is currently a research fellow at the Media Computing Group, City University of Hong Kong. His research interests include general Markov Random Fields modeling, discrete/continuous energy minimization, image segmentation, weakly-supervised clustering, structural authentication, and generic pattern recognition.

Jiaya Jia received the PhD degree in computer science from the Hong Kong University of Science and Technology in 2004. He joined the Department of Computer Science and Engineering at the Chinese University of Hong Kong in September 2004, where he is currently an assistant professor. His research interests include vision geometry, image/video editing and enhancement, image deblurring, and motion analysis. He has served on the program committees of ICCV, CVPR, ECCV, and ACCV. He is a member of the IEEE.

Zhi-Qiang Liu received the MASc degree in aerospace engineering from the Institute for Aerospace Studies at the University of Toronto and the PhD degree in electrical engineering from the University of Alberta, Canada. He is currently with the School of Creative Media, City University of Hong Kong. He has taught computer architecture, computer networks, artificial intelligence, programming languages, machine learning, pattern recognition, computer graphics, and art and technology. His interests include scuba diving, neural-fuzzy systems, painting, gardening, machine learning, mountain/beach trekking, photography, media computing, horse riding, computer vision, serving the community, and fishing.